

72A
1ej

UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO



FACULTAD DE INGENIERIA

“ SISTEMA PARA EL CONTROL DE LA
INFORMACION DE CAJEROS AUTOMATICOS
DE BANCOMER, S.A. A NIVEL NACIONAL ”

T E S I S
QUE PARA OBTENER EL TITULO DE:

Y QUE PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA

P R E S E N T A:
E. ARMANDO HERNANDEZ DIAZ

DIRECTOR DE TESIS: M. EN I. JUAN CARLOS ROA BEIZA



México, D. F.

1992

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

INDICE.

Introducción.

Capítulo I. Recopilación y análisis de información.

- I.1 Planteamiento del problema.**
 - I.1.1 Red de teleproceso.**
 - I.1.2 Arquitectura SNA.**
 - I.1.3 Enlaces de comunicaciones.**
 - I.1.4 Importancia de los cajeros automáticos.**
 - I.1.5 Funciones del Área de teleproceso.**
 - I.1.6 Planteamiento.**
- I.2 Areas involucradas en el servicio de cajeros automáticos.**
 - I.2.1 Areas relacionadas con el servicio de cajeros automáticos.**
 - I.2.2 Información requerida.**
 - I.2.3 Procedimientos usados actualmente.**
- I.3 Estandarización de formas para reportes.**
- I.4 Análisis y elección de hardware.**
- I.5 Análisis y elección de software.**

Capítulo II. Teoría de bases de datos.

II.1 Ruta crítica.

II.2 Ciclo de vida.

II.2.1 Modelo de fases.

II.2.2 Modelo de costos.

II.2.3 Modelo del prototipo.

II.2.4 Modelo de versiones sucesivas.

II.2.5 Diseño de software.

II.3 Teoría general de bases de datos.

II.3.1 Abstracción de datos.

II.3.2 Modelos de datos.

II.3.3 Restricciones de mapeo.

II.3.4 Diagrama entidad-relación.

II.4 Modelo relacional.

II.4.1 Estructura de las bases de datos relacionales.

II.4.2 Álgebra relacional.

II.5 Normalización.

II.5.1 Primera forma normal.

II.5.2 Dependencia funcional.

II.5.3 Segunda forma normal.

II.5.4 Tercera forma normal.

II.5.5 Consideraciones de almacenamiento y eficiencia.

Capítulo III. Diseño e implementación.

III.1 Método de la ruta crítica aplicado al proyecto.

III.2 Diseño de la base de datos.

III.3 Normalización de la información.

III.4 Estructura modular del programa.

III.5 Código fuente.

III.5.1 Menú principal.

III.5.2. Módulo cajeros.

III.5.3 Módulo directorio.

III.5.4 Módulo configuraciones.

III.5.5 Módulo reportes.

III.5.6 Módulo utilería.

III.6 Manual de usuario.

Conclusiones.

Glosario.

Bibliografía.

INTRODUCCION

INTRODUCCION.

Tradicionalmente las operaciones bancarias más comunes se han realizado con dinero en efectivo, pagarés o cheques. La forma de pago actual aplicada en los sistemas bancarios más avanzados está basada en un sistema de transferencia electrónica de fondos (EFTS), que consiste tan sólo en la transmisión de información hacia el centro de procesamiento.

Los avances tecnológicos en el área de la computación y la electrónica nos permiten visualizar aplicaciones en las que el usuario (cliente) interactúa directamente con la computadora. Una de las aplicaciones prácticas en este sentido es la utilización de cajeros automáticos.

La importancia del servicio de cajeros automáticos en una institución bancaria es muy importante debido a la disponibilidad, facilidad y variedad de operaciones financieras que un usuario puede realizar en cualquier momento.

Si bien es cierto que este servicio es fácil en su manejo para el usuario, también es cierto que su entorno es muy complejo.

Por un lado está el hardware : modems, controladores , equipo y enlaces de comunicaciones, procesadores centrales, equipos especiales para transmisión de datos, etc. ; y por otro lado está el software : sistema operativo, programas para comunicaciones, etc.

Todos estos componentes de hardware y software conforman lo que es una red de teleproceso, la cual está diseñada siguiendo una arquitectura determinada, en este caso SNA.

La presente tesis tiene como objetivo el desarrollo de un sistema de base de datos para controlar la información relacionada con la definición de cajeros automáticos dentro de la red nacional Bancomer S.A.

No se profundiza en el estudio de los componentes de hardware y software involucrados en la red, sino que se enfoca a los conceptos y teoría relacionados con el desarrollo de un producto de programación.

El material está organizado de la siguiente manera.

El capítulo I es un panorama general de una red de teleproceso, explicando el tipo de arquitectura en que se basa y mencionando los componentes básicos de ésta, como son enlaces y equipo de comunicaciones. Asimismo, se resalta la importancia del servicio de cajeros automáticos y la necesidad de contar con un sistema de base de datos para el manejo de la información requerida para el análisis, atención y corrección de fallas de este servicio.

En el capítulo II se presenta la teoría relacionada con las bases de datos, como son los modelos de datos, el modelo relacional y la normalización de la información.

Se incluye el ciclo de vida, en el que se mencionan las actividades requeridas para el desarrollo de un producto de programación, y el método de la ruta crítica, necesaria para la planeación de las actividades del proyecto.

En el capítulo III se aplica la teoría anterior para el diseño de la base de datos y se presenta el conjunto de programas utilizados para el manejo de la información.

Finalmente, se anexa el manual de usuario para el uso del sistema.

CAPITULO I

RECOPILACION

Y

ANALISIS

DE

INFORMACION

I.1 PLANTEAMIENTO DEL PROBLEMA.

Una red de teleproceso está formada por un conjunto de elementos que permiten la transmisión de datos desde un punto remoto hasta un centro de cómputo para el procesamiento de información. Tiene diferentes procedimientos de operación y sus principales componentes son :

Medios de Transmisión de Datos.

Hardware (Equipo)

Software (Programación)

Además, una red de teleproceso puede operar bajo diferentes tipos de arquitectura.

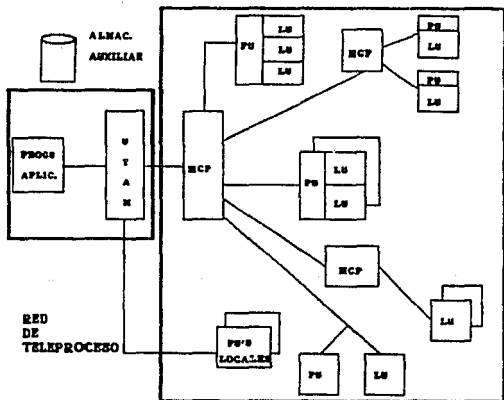
I.1.1 RED DE TELEPROCESO.

En una red de teleproceso se pretende preparar el ambiente de la misma utilizando diversos controles como apoyo. La preparación del ambiente incluye la operación de componentes de hardware como son modems, multiplexores, controladores de comunicaciones y de terminales, enlaces de comunicaciones en sus diferentes tipos y componentes de software aplicativo de comunicaciones dentro de la red.

La transmisión de la información de las aplicaciones se hace con enlaces entre las computadoras centrales empleando controladores de comunicaciones en cada uno de los centros de proceso.

Uno de los objetivos principales dentro de una red es mantener el control del funcionamiento adecuado de las aplicaciones y equipo que la constituyen, ejecutar acciones preventivas al detectar mensajes anormales o problemas con los usuarios y ejecutar acciones correctivas en caso necesario.

DIAGRAMA DE UNA RED DE TELEPROCESO.

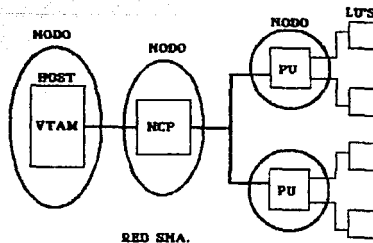


Una red de teleproceso opera bajo diferentes tipos de arquitectura. La red de teleproceso analizada opera bajo una arquitectura SNA (System Network Architecture) desarrollada por IBM y apoyada con equipo de comunicaciones como PCI, Data Stream, Motorola, Dynatech, Codex, Leedata, Timeplex, etc.

1.1.2 ARQUITECTURA SNA.

El manejo y control de una red de teleproceso con arquitectura SNA implica conocer los conceptos y terminología en los que se basa el diseño de dicha arquitectura.

SNA (System Network Architecture) es un diseño para un sistema de comunicación de datos, abarcando todas las partes de la red y que posee modos de formatos y protocolos que guían el diseño de máquinas y programas que puedan ser compatibles cuando son usados en conjunto en un sistema de procesamiento de datos. El propósito de SNA es definir formatos uniformes y protocolos para redes de comunicaciones de datos, los cuales a menudo no eran compatibles.



La arquitectura SNA está constituida por siete estratos para establecer conexión entre productos IBM.

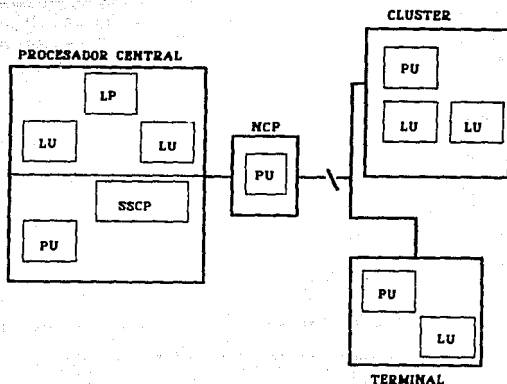
1. ADMINISTRACION DE SERVICIOS
UNIDAD LOGICA.
2. PRESENTACION.
3. CONTROL DE FLUJO
DE DATOS.
4. CONTROL DE LA TRANSMISION
5. CONTROL DE TRAYECTORIA
6. CONTROL ENLACE-DATOS (SDLC)
7. CONTROL FISICO

- 1 : Proporciona servicios para unidades direccionables de la red.
- 2 : Crea formatos para la presentación de los datos.
- 3 : Controla el flujo de datos a través de la red (cantidad)
- 4 : Controla el flujo de datos a través de la red (velocidad)
- 5 : Controla el enrutamiento de datos a través de la red.
- 6 : Controla el flujo de datos entre nodos adyacentes.
- 7 : Define la conexión física. (RS232C).

Dentro de los componentes elementales de una red SNA se tiene:

SSCP (Punto de Control para el Servicio del Sistema), es un manejador principal que tiene la capacidad de arrancar o parar la red de comunicaciones y de administrar los recursos en un dominio (subconjunto de la red). En el caso de nuestra red el SSCP equivale al manejador VTAM (Virtual Telecommunications Access Method).

VTAM (Metodo Virtual de Acceso a Telecomunicaciones), es un software que ofrece los servicios de un SSCP y reside en el procesador central (Host). Soporta componentes SNA y no-SNA, permite a los programas de aplicación compartir recursos, transfiere datos entre programas de aplicación y terminales, controla la ubicación de los recursos en la red, etc.



PU (Unidad Física), administra recursos de comunicaciones en un nodo, pudiendo ser éste una unidad de control o el código de control de terminales. En general, son las funciones de control del dispositivo donde reside y los dispositivos conectados a él.

LU (Unidad Lógica), es el puerto a través del cual un usuario final accesa a la red para comunicarse con otro usuario final. Cada LU puede soportar varios usuarios. Una LU se define también como la fuente y el destino de los datos que entran a la red, que puede ser una terminal, un cajero automático o una aplicación.

NAU (Unidad Direccionable de Red), el conjunto de LU, PU y SSCP forman los NAU's, los cuales son unidades direccionables dentro de la red.

NCP (Programa para el Control de la Red), es el sistema operativo de los controladores de comunicaciones. Descarga funciones de comunicaciones del HOST, controla a los nodos terminales, enruta datos a NCP's vecinos y controla las líneas compartidas. El NCP interactúa directamente con VTAM.

HOST (Computador Central - Huésped), controla y administra los recursos de la red. Básicamente es el procesador principal.

NODO, es un recurso de hardware o software a través del cual se establece una comunicación con uno o múltiples nodos adyacentes. Cada conjunto de programas de aplicación es considerado un nodo mayor, mientras que cada programa de aplicación individual es un nodo menor. Asimismo, un conjunto de NCP's se considera un nodo mayor y cada NCP individual es un nodo menor.

1.1.3 ENLACES DE COMUNICACIONES.

Como se mencionó anteriormente, en una red de teleproceso, la transmisión de datos desde un punto remoto hasta el centro de cómputo se realiza a través de los diferentes enlaces de comunicaciones. Es mediante estos enlaces que se tiene la conexión física y se puede establecer una sesión.

Existen diferentes medios de comunicación, entre los cuales podemos distinguir :

Enlaces vía telefónica (conmutada y privada).

Par trenzado.

Cable coaxial.

Radio VHF/UHF.

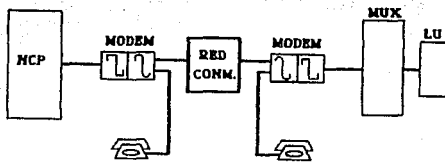
Microondas.

Fibra Óptica.

Satélite.

Enlaces Vía Telefónica.

Pueden ser conmutados o dedicados. En los enlaces conmutados la calidad del enlace es aleatoria y se hace a través de un par de alambres. En este tipo de enlaces se alcanza una velocidad de hasta 2400 bps.



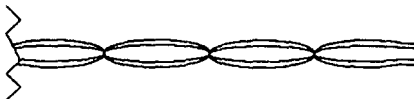
ENLACE POR LINEA CONMUTADA

Los enlaces dedicados (también llamados líneas privadas) se hacen normalmente a cuatro hilos y se pueden acondicionar para transmitir a velocidades más altas, 9600 bps.



Par trenzado.

Generalmente se utiliza en instalaciones privadas y al no estar sujeto a las operaciones de filtrado y multiplexaje de la red telefónica se obtienen tasas de transmisión relativamente altas. La atenuación depende principalmente de la frecuencia y el diámetro de los conductores. A mayor frecuencia mayor atenuación y a mayor diámetro menor atenuación.



PAR TRENZADO

Cable Coaxial.

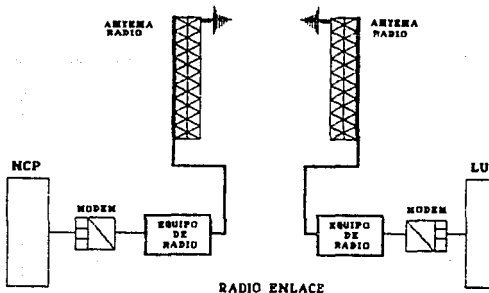
También es usado en instalaciones privadas y a diferencia del par trenzado, a altas frecuencias no presenta la misma atenuación.



CABLE COAXIAL

Radio Enlace.

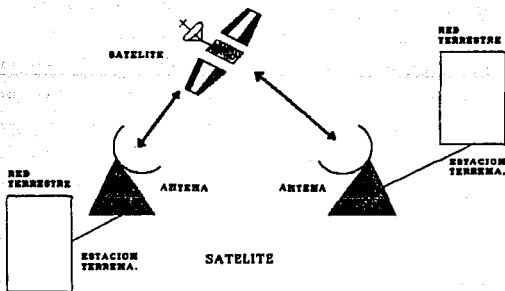
Tiene características de propagación por trayectorias múltiples en áreas urbanas. Normalmente la antena de recepción es baja (sin línea de vista a la antena emisora). La propagación es por dispersión y por reflexiones múltiples de obstáculos cercanos.



RADIO ENLACE

Satélite

Una de las principales características de los enlaces vía satélite es proporcionar comunicaciones punto-multipunto, insensibles de la distancia dentro de áreas de cobertura muy extensas. Los espectros más usados en comunicaciones vía satélite son 6.4 GHz (Banda C) con frecuencia ascendente de 5.752 a 7.075 GHz y descendente de 3.4 a 4.8 GHz. La transmisión de datos se efectúa desde una estación terrestre donde la señal en banda base es procesada y transmitida por una portadora de radio frecuencia al satélite (enlace ascendente). El satélite recibe la señal, la amplifica, la desplaza en frecuencia (para evitar interferencia) y la retransmite a la tierra (enlace descendente). La órbita de un satélite es una elipse con el centro de la tierra en un foco. La línea que une el centro de la tierra y el satélite barre áreas iguales en intervalos iguales.



I.1.4 IMPORTANCIA DE LOS CAJEROS AUTOMATICOS EN UNA RED DE TELEPROCESO PARA APLICACIONES BANCARIAS

Tradicionalmente las operaciones bancarias más comunes se han realizado con dinero en efectivo, pagarés o cheques. La forma de pago actual aplicada en los sistemas bancarios más avanzados está basada en un sistema de transferencia electrónica de fondos (EFTS), que consiste tan sólo en la transmisión de información hacia el centro de procesamiento.

Los avances tecnológicos en el área de la computación y la electrónica nos permite visualizar aplicaciones en las que el usuario (cliente) interactúa directamente con la computadora. Una de las aplicaciones prácticas en este sentido es la utilización de cajeros electrónicos.

Los cajeros electrónicos en línea han estado en operación desde 1960. En Japón, en 1965, los sistemas en línea con conexión del cajero electrónico a la computadora empezaron a ser utilizados para realizar transacciones ordinarias. En Europa se instalaron los primeros cajeros electrónicos en 1967, y para 1977 estaban en operación aproximadamente 5,000 equipos. En Estados Unidos, en 1976 existían aproximadamente 5,300 equipos, de los cuales el 90 % eran cajeros automáticos y el 10% eran despachadores de billetes.

Bancomer lanzó al mercado el servicio de cajeros automáticos en línea en el año de 1982, año en que se instalaron los primeros 34 cajeros. En el año de 1990 se contaba con 713 máquinas (322 en el Área

Metropolitana, más 391 en Banca del Interior), número que superaba la capacidad instalada en cualquier otra institución crediticia en este rubro. Para 1992, la institución cuenta con una red de 1400 cajeros automáticos ubicados en toda la República Mexicana.

Entre los principales fabricantes de cajeros automáticos se encuentran los siguientes distribuidores :

- IBM Corp.
- DIEBOLD Corp.
- BURROUGS Corp.
- NCR Corp.
- LEFEBURE Corp.
- DOCUTEL Corp.
- INCOTERM Corp.
- AEROMARINE ELECTRONICS Inc.
- MOSLER SAFE Co.
- BANK COMPUTER NETWORK Corp.

Cajero automático o electrónico CATMO

Es una máquina electromecánica que permite a los clientes (usuarios) de un banco la realización de transacciones en sus cuentas a cualquier hora del día, los 365 días del año.

Dentro de una red de teleproceso basada en una arquitectura SNA, un cajero automático o ATM se clasifica como una unidad lógica (LU) y es por medio de ésta que un usuario interactúa con la red para realizar operaciones de sus cuentas bancarias.

I.1.5 FUNCIONES DEL AREA DE TELEPROCESO EN LA ATENCION DE CAJEROS AUTOMATICOS

Debido a la importancia y gran desarrollo del servicio de cajeros automáticos dentro de nuestra red, se han establecido políticas y funciones específicas para el control y manejo eficiente del servicio que se proporciona a través de este recurso.

El Area de Teleproceso :

- Es el punto de contacto con el usuario de los recursos de la red para proporcionarle asesoría, orientación e información general sobre la solución a problemas y procedimientos operativos para la mejor utilización de la red.
- Registra en controles, tanto las interrupciones de los servicios como las fallas de los componentes de software y hardware.
- Coordina las intervenciones al equipo de comunicaciones de manera que se reduzcan los riesgos de impactar la continuidad en el servicio en línea.
- Optimiza los procedimientos de operación y solución a los problemas existentes.
- Elabora y actualiza las configuraciones de hardware y software.
- Genera listados de las sucursales con sus respectivos ATM's.
- Elabora estadísticas para la medición del servicio a usuario.

I.1.6 PLANTEAMIENTO.

La atención de cajeros automáticos (ATM's) dentro del departamento de teleproceso es una actividad prioritaria. Actualmente se cuenta con diferentes procedimientos y herramientas de software enfocados a actividades como el monitoreo de la red, y el análisis del equipo de comunicación. Sin embargo no se cuenta con los medios adecuados para el diseño de estadísticos, reportes y configuraciones de ATM's. Para lo anterior se cuenta con información impresa poco práctica en su consulta debido al gran volumen de información, ya que la forma de almacenamiento es en carpetas y la consulta no es muy ágil. Por ello y debido al constante crecimiento de la red, se necesita contar con un medio que permita la actualización constante y el manejo sencillo de toda la información requerida para la atención eficaz de los cajeros automáticos.

En base a los requerimientos del departamento, se propone desarrollar un sistema para el control de la información de cajeros automáticos. Este sistema permitirá el acceso a la información relacionada con la aplicación de cajeros automáticos, como lo es :

- 1) Unidad lógica
- 2) Unidad física.
- 3) Lineset.
- 4) Tipo de Enlace.
- 5) Equipo de comunicaciones.
- 6) Ubicación física.
- 7) Tipo de cajero.
- 8) Resolvedores de otras áreas involucradas con los ATM's.

Además contará con gráficos que nos muestren la configuración del enlace entre el cajero y el Host (procesador central). Se podrá efectuar altas, bajas, modificaciones e impresión de formatos y configuraciones de los ATM's.

Este sistema será amigable por el fácil manejo para el usuario, ya que contará con el uso de ventanas y una opción de ayuda que se podrá invocar desde cualquier nivel del sistema.

1.2 AREAS INVOLUCRADAS EN EL SERVICIO DE ATM'S

Existen diferentes áreas que intervienen directamente en el servicio de cajeros automáticos. Cada una de estas realiza funciones específicas pero que, a la vez, mantienen una relación estrecha con el fin de brindar un buen servicio. Algunas de estas áreas son: Departamento de Teleinformática, Departamento de Control de Cajeros Automáticos y el Departamento de Teleproceso.

1.2.1 AREAS INVOLUCRADAS CON EL SERVICIO DE ATM'S.

Departamento de Control de Cajeros Automáticos : vigila el servicio de ATM's en base a un monitoreo de las operaciones realizadas con este recurso y elabora estados de disponibilidad por cajero y sucursal con el objeto de dar continuidad al servicio. Asesora a los clientes-usuarios del servicio y da seguimiento junto con otros departamentos a los problemas detectados.

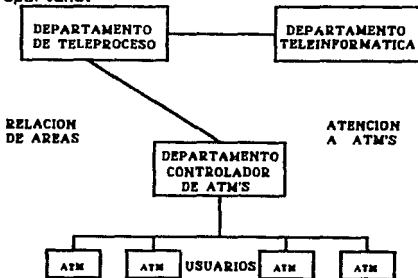
Departamento de Teleinformática : se encarga de la instalación del equipo de comunicaciones necesario para el funcionamiento del cajero automático, así como de su mantenimiento. Por otra parte, supervisa los medios de transmisión dentro de la red y realiza acciones correctivas en fallas presentadas dentro de la misma.

Departamento de Teleproceso : proporciona asesoría, orientación e información general para la solución de problemas operativos y es quien primeramente se encarga de aplicar las acciones correctivas necesarias para restablecer el servicio. Además se encarga de coordinar las intervenciones al equipo de comunicaciones de manera que se reduzcan los riesgos de impactar la continuidad en el servicio.

Estas áreas trabajan conjuntamente para la detección y corrección de fallas dentro de la red.

El punto inicial para la detección de fallas es el usuario, quien, vía telefónica, reporta al Centro Controlador de Cajeros Automáticos problemas en el servicio del cajero. Como primera instancia, este departamento asesora al usuario en el manejo del ATM y en la solución del posible problema. En el caso de que los medios con que cuenta este departamento no sean suficientes para solucionar el problema se comunica con el Área de Teleproceso, quien se encarga de verificar por medio de software el estado del recurso, además de checar el equipo de comunicaciones, el enlace y la calidad de la señal de transmisión. De ser necesario, realiza acciones correctivas al equipo. Si las fallas presentadas en el equipo o en el enlace son fallas físicas se reportan al Departamento de Teleinformática, el cual es responsable de tomar las acciones correctivas que sean necesarias.

Durante todo este proceso de detección, diagnóstico y corrección de fallas es necesario contar con medios rápidos y eficientes para la consulta de la información, de tal forma que el restablecimiento del servicio sea oportuno.

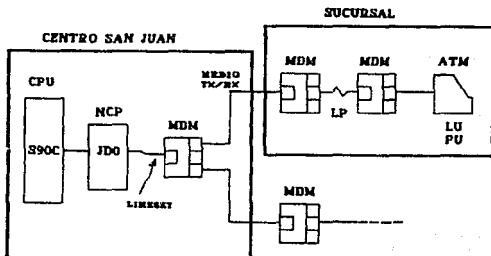


1.2.2 INFORMACIÓN REQUERIDA.

La importancia del servicio de cajeros automáticos en una institución bancaria es importante debido a su disponibilidad y a la variedad de operaciones que un usuario puede realizar en cualquier momento, lo que nos lleva a la necesidad de contar con controles eficientes de los recursos de la red que hacen posible su funcionamiento.

Se usará el término "Control" para designar un archivo en el que se encuentra información asociada con cada uno de los recursos de la red que intervienen directamente en la comunicación entre el cajero automático y el procesador central.

A continuación se ejemplifica una configuración de enlace en la cual se muestra la información manejada en un control y se explica el uso de cada dato que aparece en la configuración.



Nombre de Sucursal : este dato es necesario para conocer la ubicación física del ATM.

Número de Sucursal : se utiliza para llevar un control numérico de las sucursales.

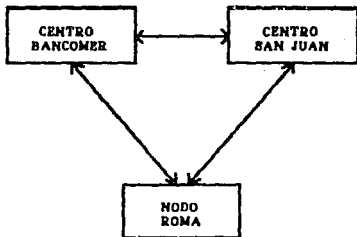
Centro de Cómputo : la red cuenta actualmente con tres grandes centros de cómputo, los cuales son :

- 1.- Centro Bancomer.
- 2.- Centro San Juan.
- 3.- Nodo Roma.

En el Nodo Roma únicamente se cuenta con equipo de comunicaciones y no se tiene un procesador central.

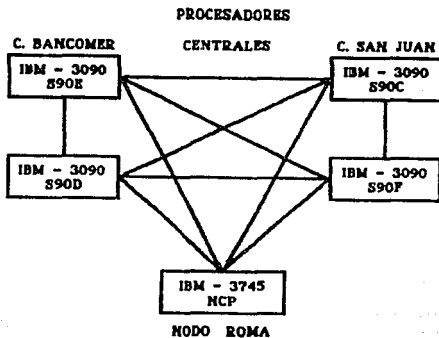
Este dato es importante para conocer en cuál de estos centros se encuentra físicamente el equipo de comunicaciones.

A continuación se muestra la interconexión entre estos centros.



CENTROS DE COMPUTO DE LA RED DE TELEPROCESO

Procesador Central : indica por medio de qué equipo se pueden realizar acciones para el diagnóstico y corrección de fallas del ATM. Los CPU's con que cuenta la red son los siguientes:



NCP : sirve para conocer la ruta por la que se establece la comunicación entre el ATM y el procesador central.

Lineset : con este dato se puede conocer el NCP al que está conectada la PU (unidad física) del recurso en cuestión, así como el estado en que se encuentran los recursos conectados a él.

PU (unidad física) : proporciona información acerca del controlador en el que se encuentra el recurso a analizar.

LU (unidad lógica) : es la identificación del recurso final (ATM) que será analizado.

Enlace : indica el medio de comunicación por el que viaja la información entre el ATM y el procesador central.

Tipo de Cajero : es el nombre del fabricante y es importante conocerlo ya que cada uno tiene diferente tipo de comportamiento.

Horario de Servicio : proporciona el número de hrs/día que la institución se compromete a dar servicio a sus usuarios. Este dato sirve para obtener estadísticas de disponibilidad por sucursal.

Centro Regional : dato manejado para ubicar el lugar geográfico en que se encuentra el cajero. Cada centro regional es un estado de la República Mexicana.

Respaldo de Enlace : es la vía alterna del enlace de comunicaciones en el caso de que se pierda el enlace primario.

Dirección y Teléfono de Sucursal : dato necesario para poder comunicarse con el personal de la sucursal y realizar pruebas en caso de fallas con los ATM's.

Directorio de Areas : Contiene información para la comunicación oportuna con áreas resoledoras en caso de problemas con ATM's.

Total de ATM's por Sucursal : proporciona información para estadísticas.

Resplado de PU : en el caso de fallas con definiciones normales de un cajero, se cuenta con respaldo alternos listos a configurar y corregir la falla.

Equipo de Transmisión y Recepción : es el equipo que interviene en la comunicación entre el NCP y el ATM.

La información anterior requiere de una constante actualización debido a que continuamente se incorporan más recursos de ATM's y los ya existentes sufren cambios en su configuración de red.

A nivel operación esta información es necesaria para la detección corrección y diagnóstico de fallas y se requiere contar con controles que contengan información completa. A nivel administrativo, el tipo de control requerido es diferente ya que la información se usa para la generación de estadísticas para la medición de servicio, generación de gráficas, etc.

I.2.3 PROCEDIMIENTOS USADOS ACTUALMENTE.

En el área de teleproceso se cuenta con diversos controles, los cuales no cumplen con los requerimientos de información y es necesario consultar varios formatos para obtener la información suficiente durante el proceso de detección, diagnóstico y corrección de fallas.

En la elaboración de dichos controles no se cuenta con un estándar a seguir ya que cada uno de ellos tiene diferente presentación, la información que incluyen no es la requerida o carecen de la misma. Asimismo, se elaboran con diferentes medios, algunos con recursos para PC y otros con los recursos del procesador central.

La manera en que se elaboran y manejan actualmente es rudimentaria, pues se usan medios no adecuados para su desarrollo. Esto implica una gran inversión de tiempo y, por lo mismo, no se actualizan constantemente.

Algunos de los formatos usados actualmente se presentan a continuación.

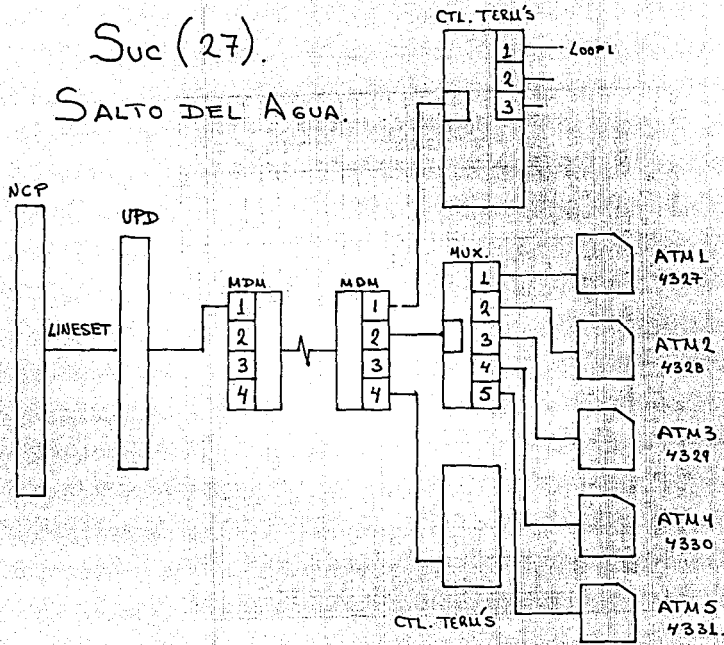
FORMATO 1.

GRUPO SISTEMAS DE INFORMACION

BANCOMER. S. N. C.

MANUAL DE VERIFICACION DEL CONTROLADOR											
CONFIGURACION DEL CONTROLADOR 4702 M.M. 099											
SUCURSAL: (044) SAN ANGEL											
D A T O S D E L S I S T E M A											
SUCURSAL SOFTWARE HARDWARE											
NO	APLC	NUM	SAFE	C I C S	V T A M	243553P31	676728C2D	0001	01	NO	P
ID		SUC		NORMAL	RESPAL	NORMAL	RESPALDO	440024 BR	P	01	LO
01						FAA20N03	FAA20R03				
03				F099	G099	LUF099	LUG099				
04	INS							X X		1	01 08
05	OCCB	044	1083	1083	6083	LU1083	LU6083	X X X		2	01 08
06	OCCB	044	1082	1082	6082	LU1082	LU6082	X X X		2	02 08
07	SESP	044	0451	0451	5451	LU0451	LUS451	X		2	03 08
08	SESP	044	0446	0446	5446	LU0446	LUS446	X		2	04 08
09	SESP	044	0443	0443	5443	LU0443	LUS443	X		2	05 08
10	SESP	044	0444	0444	5444	LU0444	LUS444	X		2	06 08
11	SESP	044	0445	0445	5445	LU0445	LUS445	X		2	07 08
12	SESP	044	0453	0453	5453	LU0453	LUS453	#		2	08 08
13	SESP	044	0450	0450	5450	LU0450	LUS450	X X		3	01 08
14	SESP	044	0448	0448	5448	LU0448	LUS448	X		3	02 08
15	SESP	044	0452	0452	5452	LU0452	LUS452	X		3	03 08
16	SESP	044	0447	0447	5447	LU0447	LUS447	X		3	04 08
17	SESP	044	2162	2162	7162	LU2162	LU7162	X		3	05 08
18	SESP	044	0449	0449	5449	LU0449	LUS449	X		3	06 08
19	SESP	044	0454	0454	5454	LU0454	LUS454	X		3	07 08
20	SESP	044	2163	2163	7163	LU2163	LU7163	X		3	08 08
35	AGER	044				LU20TP11	LU20SP11	X			00

Suc (27). SALTO DEL AGUA.



34

FORMATO 2.

En el proceso de actualización de la información se efectúan varios pasos, los cuales incluyen altas, bajas y modificaciones. En general, estos procedimientos no funcionan como debieran al no ser óptimos y no contar con una base de datos adecuada para los fines perseguidos ya que la información se almacena en diferentes archivos que no guardan ninguna relación entre sí. Cabe mencionar que éstos son puramente archivos de texto elaborados en editores y procesadores de palabra.

Cuando un ATM es instalado en la red como un nuevo recurso, el área de teleproceso requiere la información necesaria para su monitoreo, como es la ubicación física, configuraciones, y sus definiciones dentro del Host.

En el procedimiento para la alta, baja y modificación se siguen los siguientes pasos :

- 1.- Elaborar la configuración para este nuevo recurso.
- 2.- Accesar diferentes archivos (archivos de texto) para incluir esta información a través del software del Host y algunos en software de PC.
- 3.- Impresión de formatos para incluirlos en carpetas de consulta.

Como se puede observar, no se cuenta con un método que nos permita realizar las funciones anteriores de una forma rápida y eficiente.

1.3 ESTANDARIZACIÓN DE FORMAS PARA REPORTES.

Los controles que se requieren en el área de teleproceso y que contienen la información necesaria para el manejo, control y diagnóstico de un ATM no cuentan con un criterio a seguir en su elaboración, ya que cada área, e incluso cada persona, maneja la información que estima necesaria para su uso y no considera las necesidades generales. Al no existir un estándar se tiene como resultado varios formatos de presentación, que tienen deficiencias en su contenido.

Por otra parte, se cuenta con información proporcionada por otras áreas, lo cual implica una cierta dependencia en cuanto a información actualizada. Estas áreas tienen sus propios requerimientos y no se ajustan a necesidades generales, ya que esta información está incompleta o no es necesaria en algunos casos, por lo cual los formatos requieren de modificaciones en su contenido.

Se analizan 4 tipos de formatos utilizados en el análisis formal de un ATM. Al igual se revisan configuraciones básicas de representación de enlaces incluyendo equipo de comunicaciones.

CONTROL NUMERO 1.

CARACTERISTICAS :

- 1.- Nombre de la Institución Bancaria.
- 2.- Título del control.
- 3.- Número del controlador e identificación.
- 4.- Nombre y número de la sucursal.
- 5.- Número de ATM's.
- 6.- Número de loop.
- 7.- Localización de sucursal. (Area Metropolitana o C. Regional).

OBSERVACIONES.

- 1.- Dato correcto, indica el nombre de la Institución.
- 2.- El título no indica si el formato se realizó para cajeros IBM o Diebold.
- 3.- Subtítulo " Controlador " es repetitivo y redundante.
- 4.- Subtítulo " Suc " (sucursal) es repetitivo.
- 5.- Número de ATM se indica correctamente.
- 7.- El loop no es un dato necesario para el área de teleproceso.
- 8.- Subtítulo " Area Metropolitana " es repetitivo.

CONCLUSIONES :

Del análisis anterior se puede concluir que existen deficiencias en la presentación, datos inecesarios y redundantes, además de que carece de información indispensable.

CONTROL 1.

BANCOMER, S. N. C.

CENTRO: SAN JUAN.

* RELACION DE CAJEROS AUTOMATICOS DE LA RED DE TELEPROCESO *

CONTROLADOR CONV. IA. METROPOLITANA		CONTROLADOR CONV. IA. METROPOLITANA	
SUC N003 - PUJABND5	CAJEROS LOOP	SUC N004 - PUJABND1	CAJEROS LOOP
0 ISAN	4159-60 04	0	4071 04
1 ANTONIO	4864/80 04	4 FRAY BERVANDO	4875-76 04
5 ABAD	4908-89 04	2	4974 04
N000 ROMA IA. METROPOLITANA		CONTROLADOR 166 IA. METROPOLITANA	
SUC N011 - PUV26ND1	CAJEROS LOOP	SUC N012 - PUA42ND2	CAJEROS LOOP
036 CONDESA	4025-26	CIA TORRE PEMEX	4123
	4881-82		
CONTROLADOR CONV. IA. METROPOLITANA		CONTROLADOR 174 IA. METROPOLITANA	
SUC N014 - PUN40ND1	CAJEROS LOOP	SUC N015 - PUN59ND2	CAJEROS LOOP
199 CENTRAL DE ABASTOS	4147-48 04	094 LAS AGUILAS	4141
II	4879 04		
CONTROLADOR CONV. IA. METROPOLITANA		CONTROLADOR CONV. IA. METROPOLITANA	
SUC N020 - PUN55ND1	CAJEROS LOOP	SUC N022 - PUJACND5	CAJEROS LOOP
05A PARQUE INDUSTRIAL	4466 04	049 SANTA	4077/97 04
	4883-84 04	CLARA	4192-93 04
N000 ROMA IA. METROPOLITANA		CONTROLADOR 301 IA. METROPOLITANA	
SUC N023 - PUV34ND2	CAJEROS LOOP	SUC N024 - PUC05ND1	CAJEROS LOOP
022 LUCERNA	4140	027 SAN JUAN DE LETRAN	4163
CONTROLADOR CONV. IA. METROPOLITANA		CONTROLADOR CONV. IA. METROPOLITANA	
SUC N025 - PUJ18ND1	CAJEROS LOOP	SUC N026 - PUJABND5	CAJEROS LOOP
054 GLORIETA	4256-57 04	001 MERCED	4355-86 04
CUITLAHUAC	4323-24 04		4858-59 04
CONTROLADOR CONV. IA. METROPOLITANA		CONTROLADOR CONV. IA. METROPOLITANA	
SUC N027 - PUB76ND5	CAJEROS LOOP	SUC N036 - PUJ26ND4	CAJEROS LOOP
064 GRANJAS MEXICO	4528-29 04	070 SATELITE	4037 04
	4898-99 04		
CONTROLADOR CONV. IA. METROPOLITANA		CONTROLADOR CONV. IA. METROPOLITANA	
SUC N039 - PUV20ND1	CAJEROS LOOP	SUC N042 - PUP8AND5	CAJEROS LOOP
018 LOMAS	4866-67 04	024 TEXCOCO	4135/871 04
CONTROLADOR CONV. IA. METROPOLITANA		CONTROLADOR CONV. IA. METROPOLITANA	
SUC N053 - PUV26ND3	CAJEROS LOOP	SUC N055 - PUI14ND1	CAJEROS LOOP
048 IBM LEGARIA	4936 04	055 UNIV. AURRERA	4896-97 04

CONTROL NUMERO 2.

CARACTERISTICAS :

- 1.- Nombre del centro de cómputo.
- 2.- Título.
- 3.- Número del controlador e identificación.
- 4.- Nombre y número de la sucursal.
- 5.- Identificación de lineset.
- 6.- Tipo de enlace.
- 7.- Loop del ATM.

OBSERVACIONES.

- 1.- Nos indica el centro de cómputo por el cual están comunicadas dichas sucursales, pero no se indica el nombre de la Institución Bancaria.
- 2.- No existe título específico y bien definido que nos indique de qué se trata este control. El tipo de letra para el título no es el adecuado.
- 3.- El Número de controlador aquí presentado no proporciona ninguna ayuda en caso de fallas con este controlador.
- 4.- El modelo del controlador es redundante, ya que existe sólo un modelo de controlador utilizado para los ATM's en la red.
- 5.- El número de sucursales correcto, ya que es necesario en un análisis
- 6.- Identificación corecta de lineset.
- 7.- Para un diagnóstico efectivo se recurre a este dato (tipo de enlace), indispensable en un buen control.

B.- Indica en qué loop del controlador de terminales se encuentran conectados los ATM's en caso de ser IBM, pero no nos indica cuántos ni cuáles son los ATM's y además no es necesario.

CONCLUSIONES.

Se puede deducir que en este control son necesarias varias correcciones, agregarle información, así como mayor actualización. Se le incluirían los siguientes datos para complementarlo :

- a) Nombre de la Institución Bancaria
- b) Tipo y Número de ATM.
- c) Eliminar redundancia en el número de controlador.

CONTROL 2.

C O N T R O L A D O R	D O C U M E N T O	T I P O	S U C U R S A L	L I N E S	R E G I S T R O	L I N E - S E T	E M B L A C E
NUM IDENTIFICACION	NUM IDENTIFICACION						
095	PJADN05	4702	001 MERCED	4	2-3	LIJ119	RADIO
096	PUI52N01	4702	002 PARAGUAY	4	2-3	LI1056	RADIO
037	PUI59N05	4702	004 MORELOS	4	2-3	LIJ104	RADIO
195	PUI00N01	4702	007 UCLLAS ARTES	4	2-3	LIJ122	RADIO
129	PUI16N01	4702	008 DOCTORES	4	2	LI1016	LP
013	PUI59N01	4702	014 JUAREZ	4	2-3	LI1055	RADIO
106	PJABN05	4702	015 SAN ANTONIO ABAD	4	2-3	LI1117	RADIO
050	PJAFN01	4702	016 NONGALCO	4	2-3	LIJ121	RADIO
317	PUI50N01	4702	021 VILLA CONSULADO	4	2-3	LI1050	RADIO
003	PJUAN07	4702	023 CORREGIDORA	4	2	LIJ100	LP
031	PJAGN01	4702	025 CINCO DE MAYO	4	2-3-5	LIJ112	RADIO
171	PUI28N01	4702	026 LOMAS VERDES	4	2-3	LI1004	RADIO
170	PJ132N02	4702	027 SAN JUAN DE LETRAN	4	2-3	LI1032	LP
047	PJ132N03	4702	029 A. DE CIRCUNVALACION	4	2-3	LI1032	RADIO
022	PUI52N01	4702	031 SN BARTOLO NAUCALPAN	4	2-3	LI1052	RADIO
005	PJJA7N01	4702	034 VALLEJO	4	2-3	LIJ113	RADIO
052	PJIG2N01	4702	049 NI/O PERDIDO	4	2-3	LI1062	RADIO
070	PJAJ9N01	4702	042 FRAY BERNARDO	4	2-3	LIJ115	RADIO
131	PUI54N01	4702	045 TACUBA	4	2-3	LI1054	RADIO
048	PJACN05	4702	049 SANTA CLARA	4	2-3	LIJ118	RADIO
034	PJABN01	4702	050 INSURGENTE SANDALO	4	2-3	LIJ114	RADIO
010	PUI57N01	4702	051 LINDAVISTA	4	2-3	LI1057	RADIO
310	PJ118N01	4702	054 GLORICETA CUITLAHUAC	4	2-3	LI1018	RADIO
049	PJ136N01	4702	057 VILLA NORO	4	2-3	LI1036	RADIO
414	PJ590N01	4702	060 DOCTEZUMA	4	2-3-5	LIJ098	RADIO
100	PJARAN01	4702	061 INGUARAN	4	2-3	LIJ116	RADIO
108	PJ133N01	4702	063 AVILA CAMACHO	4	2-3	LI1033	RADIO
162	PJ595N01	4702	066 JAMAICA	4	2-3	LIJ099	RADIO
147	PUI26N41	4702	070 SATELITE	4	2-3	LI1026	RADIO
069	PJJD4N07	4702	074 AEROPUERTO	4	2	LIJ126	LP
033	PJ163N01	4702	076 JARDIN BALBUENA	4	2-3	LI1063	RADIO
068	PJ161N07	4702	077 VALLEJO CUITLAHUAC	4	2-3	LI1061	LP
112	PJ161N08	4702	080 SAN JUAN DE ARAGON	4	2-3	LI1061	RADIO
112	PJ161N09	4702	081 PLAZA SATELITE	4	2-3	LI1066	RADIO
315	PJ161N02	4702	083 PLAZA SATELITE	4	2-3	LI1066	LP
318	PJ139N01	4702	084 AGRICOLA ORIENTAL	4	2-3	LI1039	LP
315	PJG51N01	4702	086 CD. NETZAHUALCOYOTL	4	2-3	LI1051	RADIO
161	PJUB5N01	4702	088 VALLE DONADO	4	2-3	LI1127	RADIO

CONTROL NUMERO 3.

CARACTERÍSTICAS :

- 1.- Título.
- 2.- Ordenado por número de sucursal.
- 3.- Centro de cómputo al que está conectado el recurso.
- 4.- Tipo de ATM.
- 5.- Número e identificación del ATM.
- 6.- Tipo de enlace.

OBSERVACIONES.

- 1.- Título claro, indicando correctamente de qué se trata el control.
- 2.- Bien ordenado en número ascendente.
- 3.- Especifica correctamente a qué centro de cómputo está conectado.
- 4.- Identificación correcta del tipo de cajero (IBM o Diebold).
- 5.- Identificaciones de los ATM's bien definidas.
- 6.- En el tipo de enlace sólo se especifica si es vía radio.

CONCLUSIONES :

En general este control es de fácil manejo y contiene los datos mínimos indispensables para un análisis y diagnóstico de ATM's, existen detalles que pueden corregirse y de esta manera obtener un control más completo. Se podría incluir :

- a) Nombre de la Institución Bancaria.
- b) Especificar el tipo de enlace para cada sucursal.
- c) Incluir identificaciones de unidades físicas (PU's).

RELACION DE SUCURSALES

BANCOMER

N.M.	NOMBRE	C. BANC.	C. SJ	C. ROM.	RADIO	I	B	M	D	E	B	O	L
1	HERCINO	X			X	4904	4196	4903	4859	4195	4958		
2	PARAGUAY	X			X						4183	4184	5185 4186 4189
3	SAN COSME		X	X	X						4850	4851	
4	HOELOS	X			X	4224	4225	4240	4241				
5	INSURGENTES		X	X	X						4191	4192	4193 4194 4195
6	REFORMA FLORENCIA		X	X	X						4098	4104	4107 4108 4111
7	BELLAS ARTES	X			X						4247	4249	4327 4328 4361
8	DOCTORES	X			X								
9	TACUAYANA	X			X						4197	4198	4199 4200 4202
10	IZTAPALAPA CENTRO	X			X						4765	4989	4990
11	ATIZAPAH	X			X						4124	4127	4128
12	ANAHUAC		X	X	X						4152	4155	4164 4169 4170
13	TELANEPANTLA		X	X	X						4174	4176	4178 4180 4181
14	JUAREZ	X			X						4131	4132	4133 4134 4136
15	SAN ANTONIO ABAD	X			X	4159	4160	4864	4880	4908	4909		
16	JONDALCO	X	X		X						4153	4154	4894 4895
17	MIXCOAC	X			X						4826	4827	4828 4829 4836
18	LOHAS		X	X	X	4866	4867						
19	VALLE	X			X						4233	4234	4325 4226 4808
20	ROMA		X	X	X						4235	4236	
21	VILLA CONSULADO	X			X						4300	4301	4302 4303 4306
22	LUCERNA		X	X	X	4140							
23	CORREGIDORA	X			X								
24	TEXCOCO	X			R	4920	4921	4135	4871				
25	CINCO DE MAYO	X			X						4495	4496	4568 4569 4572
26	LOHAS VERDEG		X	X	X								
27	SAN JUAN DE LETRAN	X			X	4163							
28	GLORIETA COLON		X	X	X						4115	4116	4119 4120 4121
29	AVILLO DE CIRCUVALA		X		X								
30	GLORIETA ETIOPIA	X			X						4034	4035	4354 4306 4307
31	SAN BARTOLO		X	X	X						4113	4114	4385 4386 4445
32	COAHUILA		X	X	X						4115	4226	4887
33	GENOVA		X	X	X						4014	4020	4887
34	VALLEJO	X			X								
35	INSURGENTES SANTANDER	X			X						4838	4839	4844 4045 4846
36	CONDESA		X	X	X	4025	4026	4881	4882				
37	PORTALES	X			X						4528	4539	4591 4592 4593
38	MATIAS ROMERO	X			X						4784	4785	4794 4795 4848

CONTROL NUMERO 4.

CARACTERISTICAS :

- 1.- Título.
- 2.- Ordenado por número de sucursal.
- 3.- Nombre de sucursal.
- 4.- Identificación de unidad física(PU).
- 5.- Tipo de Enlace.
- 6.- Centro de Cómputo al que está conectado.
- 7.- NCP al cual está conectada la Sucursal.
- 8.- Tipo e identificación del ATM.

OBSERVACIONES.

- 1.- Título bien indicado.
- 2.- Ordenado correctamente en orden ascendente.
- 3.- La sucursal se identifica claramente.
- 4.- Identificación correcta de PU para todas las sucursales.
- 5.- No especifica los tipos de enlace para cada sucursal.
- 6.- Centro de cómputo al que se enlaza está bien identificado.
- 7.- Identificación correcta de NCP.
- 8.- Tipo e identificación del ATM se indica correctamente.

CONCLUSIONES :

Este control contiene casi en su totalidad las características y datos necesarios para una consulta fácil y completa, únicamente se puede mejorar detalles en la presentación e incluir datos complementarios. En general es un control completo.

CONTROL 4.

BANCONEER									
RELACION DE SUCURSALES Y CAJEROS									
NUM	SUCURSAL	CTL	ENL UNICO	ENL RADIO	INDO ROMA	SUC	SUC	CAJEROS EN LA SUCURSAL	
001	MERCEO	PUJADN05	*	*			JDD	IBM 4859/903-04	
002	PARAGUAY	PUISAN01	*	*			JSD	DIE 4183-86 DIE 4189-90	
003	SAN COSME	PUIV6M02	*	*		RO1		DIE 850-51	
004	MORELOS	PUJ9EW05	***	***			JDD	IBM 4224-25/40/43	
005	INSURGENTES	PUIV7N30	*	*		RO1		DIE 4101-95	
006	REFORMA FLORENCIA	PUIV8N30	*	*		RO1		DIE 4098/104 DIE 4107-08/11-12	
007	BELLAS ARTES	PUJ8DN01	*	*			JDD	DIE 4274-49 DIE 4127-28/41	
008	DOCTORES	PUIIGN04	*	*			JSD		
009	TACUBAYA	PUM28N01	*	*		S90		DIE 4110-99 DIE 4200/02/04	
010	IZTAPALAPA CENTRO	PUM35N07	*	*		S90		DIE 4765/087-90	
011	ATIZAPAN	PUP63N05	***	***		D70		DIE 4124/27-28	
012	ANAHUAC	PUM52N30	*	*		S90		DIE 4154/55 DIE 4167-68-70/72	
013	TLALNEPANTLA	PUM8DN41	*	*		RO2		DIE 4174/76/78 DIE 4180-82	
014	JUAREZ	PUI59N01	*	*			JSD	DIE 4131-34 DIE 4126-28	
015	SAN ANTONIO ADAD	PUJABN05	*	*			JDD	IBM 4159-60/4864 IBM 4880/4908-09	
016	HONDALCO	PUJAFN01	*	*			JDD	DIE 4153-54/89-95	
017	MIXCOAC	PUM30N06	*	*		S90		DIE 4826-29 DIE 4836-37	
018	LOMAS	PUV20N21	***	***		RO1		IBM 4866-67	
019	VALLE	PUN35N08	*	*		S70		DIE 4233-34/325 DIE 4326/808-89	
020	ROMA	PUM61N43	*	*		RO2		DIE 4235-36	
021	VILLA CONSULADO	PUI50N01	*	*			JSD	DIE 4300-05	
022	LUCERNA	PUM21N31	*	*		RO1		IBM 4140	
023	CORREGIDORA	PUJ9AN07	*	*			JDD		
024	TEXCOCO	PUPR8N05	*	*		D70		IBM 4135/871 IBM 4920-21	
025	CINCO DE MAYO	PUJAGN01	*	*			JDD	DIE 4495-96 DIE 4568-69/72	
026	LOMAS VERDES	PUI28N03	*	*		JSD		JSD	
027	SAN JUAN DE LETRAN	PUI32N02	*	*			JSD	IBM 4160	
028	GLORIETA COLON	PUV23N01	*	*		RO1		DIE 4115-16/19-22	
029	ANILLO DE CIRCUNV.	PUI32N03	*	*			JSD		
030	GLORIETA ETIOPIA	PUM29N01	*	*		S90		DIE 4034-35 DIE 4306-08/54	
031	SAN BARTOLO NAUCALPAN	PUI52N01	*	*		JSD		JSD	DIE 4113-14 DIE 4385-86/445
032	CDAHUILA	PUV24N30	*	*		RO1		DIE 4125-26/887	

CONFIGURACIONES DE ENLACES
Y
EQUIPO DE COMUNICACIONES

CONFIGURACION NUMERO 1.

CARACTERISTICAS :

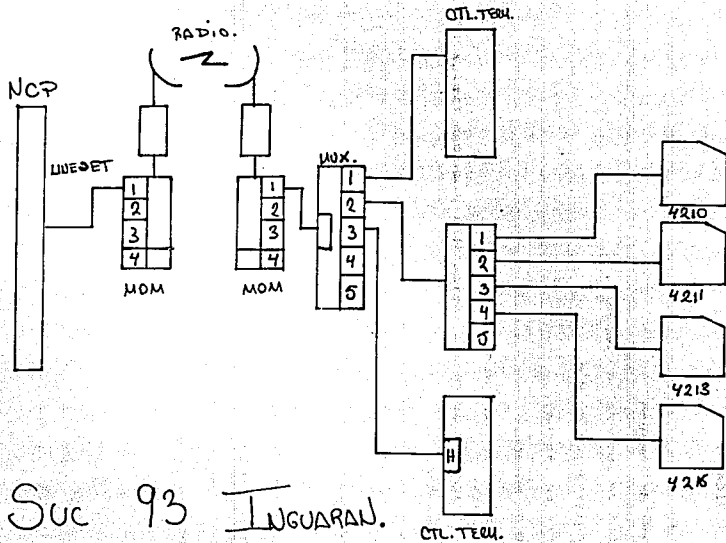
- 1.- Título.
- 2.- Nombre y número de sucursal.
- 4.- Identificación de NCP.
- 5.- Nombre de lineset.
- 6.- Tipo de enlace.
- 7.- Equipo de comunicaciones.
- 8.- Identificación de unidad lógica (LU) y unidad física (PU).

CONCLUSIONES :

Este control cuenta con deficiencias en cuanto a la presentación de la información, incluye casi en su totalidad los datos necesarios para el análisis de un ATM, se pueden realizar modificaciones e incluir más información como, por ejemplo :

- a) Nombre de la Institución Bancaria.
- b) Nombre del centro de cómputo
- c) Tipo de ATM.
- d) Estar realizando totalmente en un procesador gráfico.

Al agregar estos datos se puede contar con un control completo y eficiente para las áreas involucradas dentro del servicio de ATM's.



Suc 93 INGUARAN.

CONFIGURACION 1.

CONFIGURACION NUMERO 2.

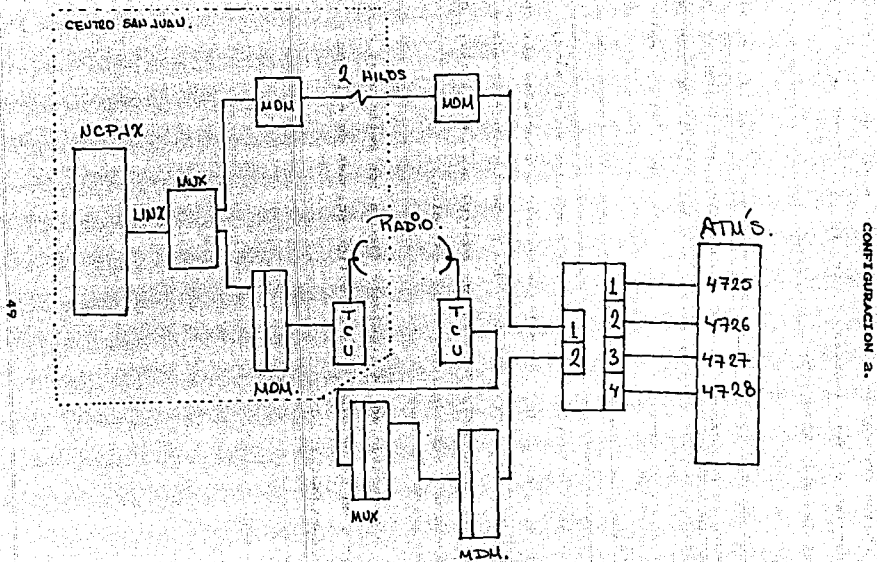
CARACTERISTICAS :

- 1.- Título.
- 2.- Centro de cómputo.
- 3.- Nombre de sucursal o compañía.
- 4.- Identificación de NCP.
- 5.- Nombre de lineset.
- 6.- Tipo de enlace.
- 7.- Equipo de comunicaciones.
- 8.- Identificación de unidad lógica (LU) y unidad física (PU).
- 9.- Tipo de ATM.

OBSERVACIONES :

Dentro de este control se cuenta con los elementos necesarios para una buena consulta que ayude al análisis de un ATM. Sin embargo, su presentación se puede mejorar si se le efectúan algunas modificaciones y se sigue un estándar para todas las configuraciones.

CONFIGURACIÓN VALLE



En los controles y configuraciones analizados se notan ciertas diferencias y deficiencias tanto en la presentación como en el contenido de información.

En algunos controles la información es insuficiente, mientras que en otros es redundante y su elaboración no contempla estándares, teniendo como resultado diversas presentaciones ya que algunos se realizan en paquetes de PC y otros utilizando software del computador central.

Debido a la importancia de los controles en el manejo de ATM's, se tiene la necesidad de realizar controles completos que contengan la información necesaria, así como una presentación adecuada y un formato específico independientemente del control que se trate, o sea, es necesario una estandarización de controles.

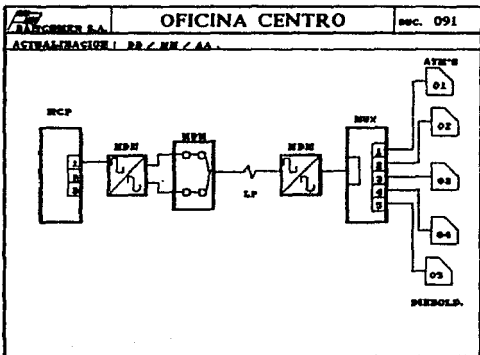
Se presentan en la siguiente figura los estándares para controles de ATM's en sucursales y para configuraciones de enlaces y equipo de comunicaciones.

ESTANDAR 1.

CENTRO DE COMPUTO SAN JUAN,
LISTADO DE CAJEROS POR NUMERO DE SUCCURSAL,
03/20/92

001 MERCED	ENLACE: RADIO	LSET: LIJ119	NCP: JDO	ATMS: IBM
4395 PUJADN05				
4396 PUJADN06				
4858 PUJADN07				
4859 PUJADN08				
4903 PUJADN09				
4904 PUJADN10				
				TOTAL: 6 CAJEROS
002 PARAGUAY	ENLACE: RADIO	LSET: LI1058	NCP: JSO	ATMS: DIEBOLD
4183 PU141N83				
4186 PU141N86				
4189 PU141N89				
4190 PU141N90				
4284 PU142N84				
4285 PU142N85				
				TOTAL: 6 CAJEROS
003 SAN COSME	ENLACE: RADIO	LSET: LI1016	NCP: ROI	ATMS: DIEBOLD
4850 PU148N50				
4851 PU148N51				
4950 PU159N50				
				TOTAL: 3 CAJEROS
004 MORELOS	ENLACE: RADIO	LSET: LIJ104	NCP: JDO	ATMS: IBM
4224 PU142N24				
4225 PU142N25				
4240 PU142N40				
4243 PU142N43				
				TOTAL: 4 CAJEROS
005 INSURGENTES	ENLACE: RADIO	LSET: LI1017	NCP: ROI	ATMS: DIEBOLD
4191 PU141N91				
4192 PU141N92				
4193 PU141N93				
4194 PU141N94				
4195 PU141N95				
				TOTAL: 5 CAJEROS
006 SF FLORENCIA	ENLACE: RADIO	LSET: LI1018	NCP: ROI	ATMS: DIEBOLD
4098 PU140N98				
4104 PU141N04				
4107 PU141N07				
4108 PU141N08				
4111 PU141N11				
4112 PU141N12				
				TOTAL: 6 CAJEROS

ESTANDAR 2



I.4 ANALISIS Y ELECCION DE HARDWARE

El hardware de alta velocidad es una forma de mejorar el desempeño de un sistema, ya que, para algunas personas, la prueba principal para determinar su eleccion es la velocidad a la que la computadora ejecuta el programa.

Para otro grupo de usuarios, que trabajan principalmente con dibujos, imágenes y fotografías en pantalla, los factores más importantes son tanto la velocidad como la resolución de la pantalla o monitor.

De la misma manera, se debe conocer con precision las necesidades para elegir un sistema operativo que se ajuste a ellas.

Las características de la máquina son objeto de un estudio, en el que se tendrá que tomar una decisión básica: contar con una computadora con las mejores características a un costo razonable.

ANALISIS DEL HARDWARE DISPONIBLE.

Las PC's cuentan con diferentes tipo de microprocesadores. Algunas utilizan procesadores 8088 - 80286 y las más recientes 80386 y 80486. Las computadoras que cuentan con el procesador 8088 tienden a ser las más baratas del mercado, pero también suelen ser mucho más lentas que las máquinas nuevas. La industria las considera pasadas de moda y ya no invierten mucho tiempo desarrollando aplicaciones para para ellas.

La 80286, sigue siendo la elección más popular y se puede utilizar para aplicaciones pequeñas a un precio razonable y que contiene todo lo necesario para un buen aprovechamiento de los recursos. Debido al enorme desarrollo de software y al requerimiento de mayor velocidad en el procesamiento de datos, se ha desarrollado el procesador 80386sx en el cual corre la mayoría del nuevo software a una velocidad mucho mayor.

El manejo de gráficas en computadora es cada vez más común. Se requiere más espacio para almacenar gráficas, textos y números. Una unidad de disco duro de 20 MB solía considerarse adecuada, pero los programas actuales requieren de mayor espacio en disco.

La memoria RAM de la computadora es otro componente en el que se debe pensar, ya que permite alojar programas de mayor tamaño.

ACER AMERICA ACER 1100/33

El sistema ACER/386, valorado en 4,365 dólares, es una de las PC's de más alto costo en el mercado en el rango de 33 MHz. Es un sistema con componentes valiosos que lo hace muy confiable.

Uno de esos componentes es el máximo de 32K de RAM para el caché del procesador. A diferencia de otros fabricantes de equipo, Acer ha sido capaz de producir un rendimiento igual o mejor que el de la competencia, aunque usando sólo la mitad de caché de otros sistemas de alto rendimiento. 32K de RAM estática a 25 ns. son suficientes para colocar la Acer como una de las mejores en cuanto a ejecución.

La unidad tiene las siguientes características :

- 32 KB de memoria caché para el procesador.
- Procesador 386sx
- Disco duro de 100 MB.
- 6 MB de memoria RAM expandible a 8 MB.
- Velocidad de 33 MHz
- Unidad de disco flexible de 5 1/4" a 1.2 MB.
- Unidad de disco flexible de 3 1/2" a 1.44 MB.
- Monitor de 14" Super VGA
- 2 puertos Seriales y un puerto Paralelo.
- Incluye Microsoft Dos 4.01
- Tarjeta de video seven fastwrite de 16 bits.

La tarjeta madre acepta hasta 8 MB de RAM, así como 16 mb adicionales en una tarjeta exclusiva de memoria que se inserta en una ranura de 32 bits.

Las velocidades disponibles en el procesador son 33 MHz, 8 MHz e "inteligente", la cual desacelera el sistema durante accesos a discos flexibles por razones de compatibilidad.

AT&T COMPUTER SYSTEMS.

AT&T 6386E/33.

El Sistema AT&T de alto rendimiento está valorado en 4,095 dólares. Este equipo cuenta con una tarjeta madre con una ranura de 8 bits, siete de 16 y dos de 32.

La unidad evaluada tiene las siguientes características:

- Procesador Intel 80386
- RAM de 4 MB de alta velocidad, expandible a 64 MB
- 64KB de memoria RAM cache
- Unidad de disco flexible de 5 1/4" a 1.2 MB.
- Unidad de disco flexible de 3 1/2" a 1.44 MB.
- Disco duro de 145 MB
- Monitor Super VBA de 14 "
- Velocidad de 33 MHz
- 2 puertos seriales y un puerto paralelo
- 124 teclado.
- Mouse Microsoft.

Concluyendo, la AT&T 6386E/33 es potente y preparada para redes y sistemas operativos alternos.

DATAWORLD DATA 386-33

Esta maquina tiene un costo aproximado de 3,980 dólares; Cuenta con una tarjeta E/S con suficiente RAM como para satisfacer a la mayoría de las computadoras. Incluye una tarjeta controladora de discos con 4MB de RAM de 80 ns. La tarjeta es tan sofisticada que no sólo tiene su propio BIOS, como tienen todos los controladores de discos, si no que también necesita configurarse para el disco que se use.

La configuración se hace a través de software cargado externamente que se incluye con el sistema. El efecto neto de este subsistema de disco es que la DATA 386-33 tiene un magnifico sistema de flujo de información.

En la prueba de procesamiento se coloca en el medio del equipo analizado gracias al RAM del sistema con velocidad de 80 ns respaldado por un ultra-rápido RAM de cache del procesador a 25 ns.

La unidad evaluada tiene las siguientes características :

- Procesador Intel 386-25
- 4 MB de memoria RAM expandible a 16 MB
- 64KB de memoria Cache
- Disco duro de 200 MB
- Unidad de disco flexible de 5 1/4" a 1.2 MB.
- Unidad de disco flexible de 3 1/2" a 1.44 MB.
- Monitor Super VGA de 14 " 1024 x 768

- 2 puertos seriales, un puerto paralelo y un puerto para juegos.
- Reloj, calendario y batería de backup.

ZEOS INTERNATIONAL 386/33

La Zeos 386/33, con 4 MB de RAM, un disco duro de 120MB y monitor VGA a color cuesta 3.882 dólares, incluyendo MS-DOS 4.01 y los propios programas de Zeos.

Los resultados de las pruebas de acceso a archivos de DOS (estructuras pequeñas) y búsquedas en disco usando el BIDS fueron mejores que el promedio debido al controlador de manejador de cache de 512K.

La unidad valuada ofrece las siguientes características :

- 4 MB de memoria RAM a 16 MB
- 64k de memoria Cache
- Disco duro de 120 a 19 ns
- Tarjeta de video 16 bits SVGA, 1 MB de video RAM
- Unidad de disco flexible de 5 1/4" a 1.2 MB
- Unidad de disco flexible de 3 1/2" a 1.44 MB
- Monitor Super VGA de 14" 1024 x 768
- 2 puertos seriales, un puerto paralelo y un puerto para juegos
- Mouse de alta resolución.

Equipada con un controlador de disco con cache la máquina Zeos es práctica, bien como una unidad individual o como un servidor de archivos de red para sistemas de aplicaciones con intenso acceso al disco.

IBM PERSONAL SYSTEM/2 MODELO 70.

El IBM Sistema Personal es un sistema de sobremesa compacto de gran rendimiento. Ofrece características avanzadas de diseño de hardware y es compatible con la mayoría de software.

El Sistema Personal de IBM Modelo 70, valorado en unos 3,500 dólares, es una PC que ofrece alta calidad en sus componentes. Con velocidad de 33 MHz, el equipo cuenta con procesador 386 sx de Intel.

IBM proporciona memoria RAM de 4 MB, capaz de producir un rendimiento de acceso a disco duro de 20 ns, soporta video SVGA de alta resolución.

La unidad valuada tiene las siguientes características :

- Procesador 386sx
- Disco duro de 100 MB.
- 4 MB de memoria RAM.
- Velocidad de 33 MHz
- 1.4 MB en disco flexible de 3 1/2"
- Monitor de 14" Super VGA
- 2 puertos seriales y un puerto paralelo

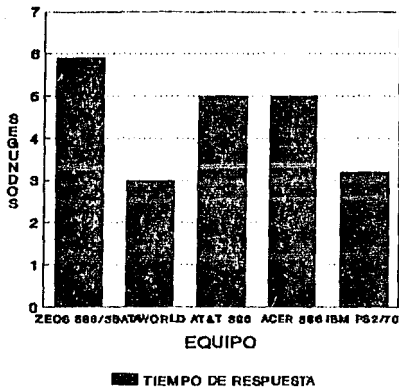
El Sistema Personal de IBM la coloca entre las mejores PC's disponibles debido a su calidad y confiabilidad en el servicio.

PRUEBAS DE EJECUCION PC BASADOS EN 386 A 33 Mhz.

ACCESO A ARCHIVO DOS. (estructuras grandes)

Mide el rendimiento del disco como resultado de la velocidad mecánica, la función del controlador de disco duro y la velocidad del bus. Tiempos menores favorecen la carga de archivos grandes.

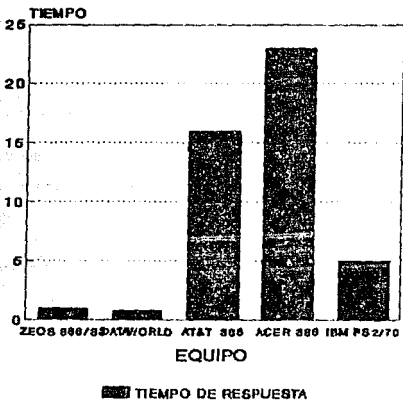
**EQUIPO PC
ACCESO A ARCHIVO DOS**



BUSQUEDA EN DISCO USANDO EL BIOS.

Mide el tiempo de acceso mecánico de pista a pista de la unidad de disco. Tiempos menores ayudan a programas tales como bases de datos, los cuales con frecuencia guardan y luego tienen que encontrar datos en diferentes lugares en un disco.

EQUIPO PC BUSQUEDA USANDO EL BIOS



SISTEMA 88

En años recientes ha ido en aumento el requerimiento para el procesamiento continuo de datos, es decir, aprovechar las 24 horas del día, los 365 días del año.

El sistema/88 fue diseñado principalmente para el procesamiento de transacciones en línea. Su diseño contra fallas y su versatilidad lo hacen idóneo para las aplicaciones en punto de venta, así como para el procesamiento de transacciones en cajeros automáticos.

CARACTERISTICAS DE ALTA DISPONIBILIDAD

TOLERANCIA A FALLAS

El Sistema/88 utiliza hardware avanzado para lograr la tolerancia a fallas. Si un puerto falla, el sistema mismo lo pone fuera de servicio y continúa su operación sin necesidad de reprogramar absolutamente nada.

CONFIGURACION DUPLICADA

Los componentes principales del sistema/88 se encuentran en una configuración duplicada. Si un componente falla su compañero duplicado continúa el procesamiento de datos.

INTEGRIDAD DE DATOS

La combinación de los componentes duplicados y del software del Sistema/88 ayudan a cumplir con la integridad de los datos.

REPARACION DINAMICA

El Sistema/88 permite reemplazos sin interrupción en la operación del sistema.

RECONFIGURACION DINAMICA

La reconfiguración dinámica resuelve el problema de tener que parar el sistema para reemplazar un puerto ó para agregarle un nuevo dispositivo.

ALTA DISPONIBILIDAD

El Sistema/88 está diseñado para operar en modo non-stop, lo cual es alcanzado a través del siguiente hardware duplicado:

- Memoria principal
- CPU
- Controladores de E/S
- Bus de datos
- Fuentes de suministro
- Discos

Está compuesto de varios microprocesadores independientes, que le proporcionan al usuario un computador de multiprocesamiento y memoria compartida.

El sistema integrado proporciona operación de disco fault tolerance a través de los siguientes dos mecanismos.

- Controladores de disco duplicados
- Espejo de archivos en disco

SELECCION DEL EQUIPO

El SBB, por su complejidad y por estar diseñado especialmente para transacciones en línea y grandes volúmenes de operaciones tanto a nivel remoto (centros regionales) como a nivel local (Área Metropolitana), así como por costo y recursos, únicamente se aplica a grandes volúmenes de información, por lo que el SBB no es apropiado para el desarrollo del sistema planteado.

El rendimiento es un factor muy importante al decidir que máquina se ajusta a las necesidades y requerimientos del sistema a desarrollar.

De las cinco máquinas PCs basadas en 386 a 33MHz, la Acer 1100/33 se diferencia por ofrecer una alta velocidad de procesador y de memoria. Construcción duradera y un diseño concienzudo se muestra tanto en Acer como IBM.

La AT&T 638E/33 Work Group System, es un servidor LAN, ofreciendo 8 cavidades de disco y 8 ranuras disponibles. Este equipo es definitivamente una elección viable para redes locales.

La Dataworld data 386-33 y la Zeos 386/33 se distinguen por su rápida ejecución de disco, gracias a controladores de disco opcionales, manejando discos duros de hasta 300MB.

Dataworld también añadió la calidad del monitor, mientras que la PS/2 de IBM se distingue por su alta resolución en el monitor VGA.

Analizando cualidades, precio y servicio, así como compatibilidad en el equipo utilizado dentro de Bancomer, el equipo que nos ofrece mayores ventajas en cuanto a velocidad de acceso al disco, resolución a gráficos y disponibilidad para su adquisición, se ha decidido utilizar el Sistema Personal Dos Modelo 70 de IBM, tanto para el desarrollo del sistema como para su instalación dentro del Área de Teleproceso.

1.5 ANALISIS Y ELECCION DE SOFTWARE PARA BASE DE DATOS Y MANEJO DE VENTANAS

CONCEPTO DE HOJA ELECTRONICA

La hoja electronica es un conjunto de programas que nos ofrece los medios para la manipulacion de datos y sirve para hacer modelos de planes y pronosticos financieros, estadisticas, estados financieros tales como: balance general, flujo de cajas, estado de resultados, etc.

Caracteristicas de la hoja electronica.

La hoja electronica es una forma de dos dimensiones, constituida por columnas y renglones. La interseccion de columnas y renglones forman celdas. En la hoja electronica se puede guardar, borrar o reacomodar la informacion.

Celda

Una celda es la interseccion de una columna y un renglón de la hoja electronica. Cada celda se identifica por las coordenadas de la intersección de la letra de la columna y el número de renglón.

Una celda puede contener un valor numerico, una formula, un texto o bien estar en blanco. El valor de la celda es igual al número que

contiene o bien al resultado del cálculo de la fórmula contenida en ella.

Ventana

Una ventana es una porción de la hoja electrónica consistente de 8 columnas y 20 renglones, y es la parte de la hoja electrónica que se visualiza en la pantalla.

LOTUS 123.

LOTUS 123 es un paquete de software integrado, que haciendo uso de su nombre cuenta con 3 aplicaciones:

- 1.- Hoja electrónica de datos
- 2.- Gráficas
- 3.- Base de datos

Como hoja electrónica de cálculo permite:

- a) Despliegue completo de los menús de acceso al sistema.
- b) Descripción de cada comando en la línea siguiente al menú.
- c) Nombre de comandos de operación familiares al usuario.
- d) Una clara distinción de comandos globales y mandos que sólo afectan a una celda o grupo de celdas.

- e) Pantalla completa de ayuda en línea relacionada con la función u operación gráfica en uso.

Como generador de Gráficos permite:

- a) Generar gráficos de línea, de barra, circulares, barras apiladas y gráficas de punto.
- b) Almacenar datos gráficos y formatos en el disco junto con la hoja electrónica.
- c) Almacenar imágenes gráficas como archivos separados.
- d) Generar nuevas gráficas al cambiar la información de la hoja electrónica.

Como manejador de Base de Datos permite:

- a) Introducir datos de registros directamente de la hoja electrónica por filas o columnas.
- b) Manejar hasta 2047 registros y 256 campos, dependiendo de la memoria del computador.
- c) Manipular los registros, incluyendo búsqueda por criterio, reorganización, generación de tablas, distribuciones y análisis estadístico.

d) Usar información de la base de datos en cálculos de la hoja electrónica, en gráficos o en ambas.

Características adicionales:

- 1) Velocidad de ejecución: una de las más rápidas hojas de cálculo disponibles en el mercado.
- 2) Control de impresión: la hoja de trabajo puede ser modificada y preparada antes de la impresión.
- 3) Intercambio de información con otros programas.

dBASE III PLUS.

Aunque "Gestión de Base de Datos" es un término informativo también se puede aplicar a las distintas formas en que se clasifica, se almacena y se usa la información. En el corazón de todo sistema de gestión de información existe una base de datos.

El lenguaje de programación dBASE III PLUS es un conjunto de órdenes que permiten acceder a la base de datos. Las órdenes del lenguaje de programación ofrecen multitud de posibilidades de manipulación de la información.

El dBASE III PLUS puede manejar 1.000 millones de registros y hasta 128 campos de registro. Cada registro puede contener hasta 4.000 caracteres. Puede ordenar varios campos al mismo tiempo y trabajar con hasta 10 archivos. Estas características tan poderosas son más impresionantes que realista. Unos simples cálculos indican que si una base de datos contiene 1.000 millones de registros de 128 campos cada uno, a una PC le llevaría más de un mes en leer una sola vez la base de datos y dicha base de datos ocuparía 30 millones de Mbytes. Pero por otro lado, el dBASE III PLUS ordena muy rápido.,

El dBASE III PLUS permite el acceso de múltiples usuarios al programa cuando se trabaja en una red de área local. Las posibilidades de protección de datos de que dispone el programa se pueden usar con las más populares redes de área local para el IBM PC y compatibles. Las limitaciones de dBASE III PLUS son pocas pero existen y una de ellas es que tiene que leer información del disco, lo que hace disminuir la velocidad de sus programas.

Para usar el dBASE III PLUS se necesita una computadora de 16 bits con MS DOS o PC DOS. la computadora debe tener un mínimo de 256K, siendo recomendable disponer de 320K y puede usar cualquier impresora de 80 columnas como mínimo.

CLIPPER

El CLIPPER es un sistema que contiene muchas facilidades para el desarrollo de aplicaciones. Se pueden crear y compilar programas sin ninguna ayuda adicional de software, tan sólo se necesita un editor de textos. Después que los programas CLIPPER han sido ligados y compilados, el archivo ejecutable puede operar con el PC/MS-DOS.

Con los programas compilados en CLIPPER no se tiene la limitante de tener que correr el programa en la computadora donde fue originalmente creado.

CLIPPER es un compilador profesional que acelera de manera considerable la ejecución de programas escritos en dBASE, además de proveer de un importante conjunto de mandatos y funciones complementarios que han hecho de él una herramienta muy útil.

Capacidad del sistema

Número máximo de registros por base de datos	1,000,000,000
Número máximo de caracteres por registro.....	RAM disponible
Número máximo de campos por registro.....	RAM disponible
Número de variables de memoria.....	2,048

Requerimientos de Hardware para usar CLIPPER.

Ordenador IBM PC, XT, AT, 386 o compatible.

Memoria mínima en RAM de 256 Kb.

1 Disco Flexible.

1 Disco Duro.

Si el equipo posee coprocesador aritmético (8087/80287) CLIPPER lo aprovecha de forma automática.

En cuanto a la red local para sistemas multiusuario, CLIPPER funciona en cualquier LAN que pueda trabajar con MS-DOS.

La versión del sistema operativo requerida es MS-DOS o PC-DOS 2.0 o superior para la versión monousuario y 3.1 o superior para la multiusuario.

SQL.

SQL es un lenguaje avanzado para el manejo de base de datos relacionales que maneja información como conjuntos lógicos llamados "relaciones" (tablas).

Es un lenguaje que puede ser utilizado interactivamente o a través de una aplicación. SQL provee un pequeño y conciso conjunto de comandos que permiten definir, desplegar y actualizar información en tablas.

La importancia real del SQL consiste en el hecho de que se ha convertido en el estándar del lenguaje para el manejo de base de datos y no en la habilidad o capacidad de servir como un lenguaje de preguntas para usuario final. En este sentido existen algunas utilerías o programas de servicio que permiten seleccionar información y condiciones por medio de un menú.

SQL fundamenta su operación en la teoría de conjuntos, lo que lo convierte en un lenguaje muy interesante y útil.

SQL está constituido de tres partes fundamentales:

- 1) Lenguaje de manipulación de datos (DML)
- 2) Lenguaje de definición de datos (DDL)
- 3) Lenguaje de control de datos (DCL)

La sección de DML es utilizada para obtener, actualizar, insertar y borrar renglones en una tabla.

La sección de DDL es usada para crear objetos de bases de datos, como índices. La sección de DCL proporciona la seguridad de la base de datos.

Dentro de los componentes de SQL existe otra sección (secundaria) llamada Sistema de Catálogo que almacena toda la información sobre los objetos de la base de datos creados usando la DDL. Las tablas de catálogo pueden, inclusive, ser modificadas utilizando la DML.

SQL cuenta también con un optimizador para seleccionar automáticamente la mejor ruta de acceso a la información. De este modo los índices no son especificados directamente, sino que el optimizador analiza el índice y selecciona el mejor índice que necesita.

SQL es un lenguaje que permite manipular estructuras de información creando tablas e índices; obtener y manipular información a través de un lenguaje de consulta; controlar el acceso a la información autorizando niveles de permiso de acceso, etc.

SQL no es un lenguaje de procedimientos con las características de interfases; provee únicamente primitivas para la estandarización de la salida de información basándose en otros productos para estos propósitos.

SELECCION DEL SOFTWARE.

De los cuatro paquetes disponibles y analizados para el manejo y manipulación de las bases de datos, creemos que CLIPPER es un sistema completo que cumple con los requerimientos necesarios para el desarrollo de nuestro sistema: facilidades para crear, compilar y ejecutar programas fácilmente.

LOTUS 123 principalmente es un paquete para el manejo de hoja electrónica. Aunque cuenta con otras aplicaciones tales, como gráficas y base de datos, no se considera apropiado para el desarrollo de nuestra aplicación debido a sus limitantes para el proceso de datos en cuanto al manejo de bases de datos.

DBASE III PLUS es un manejador de base de datos y aunque cuenta con muchas características propias y adecuadas para el desarrollo de nuestra aplicación, tiene una considerable desventaja y es que dBASE III PLUS tiene que leer información en disco lo que lo hace disminuir la velocidad de sus programas. Por otra parte, prácticamente todas las aplicaciones hechas en dBase pueden ser desarrolladas en CLIPPER, con la ventaja de hacer más rápida la ejecución del programa y tener los mismos requerimientos de Hardware.

SQL es un buen manejador de bases de datos, pero, aun cuando existe una versión (simulador) para el equipo PC, consideramos que no tiene las facilidades adecuadas para el desarrollo de la aplicación.

Por lo anterior, se ha decidido desarrollar la aplicación en CLIPPER.

CAPITULO II

TEORIA

DE

BASES

DE

DATOS

II.1 RUTA CRITICA

Prácticamente cualquier empresa se enfrenta con el problema de planear, organizar y controlar proyectos a gran escala y que, generalmente, se realizan una sola vez. El éxito del proyecto depende de muchos factores, uno de los cuales es la información que se puede obtener de las actividades que lo componen. Esta información permite responder algunas preguntas como son :

- Cuando se terminará el proyecto.
- Cuáles son las actividades que más influyen en la terminación del proyecto.
- Cuando se pueden comenzar, lo más pronto posible, y terminar, en el menor tiempo posible, las actividades del proyecto.
- Cuál es el último tiempo en el que se pueden comenzar y terminar las actividades sin que se retrase el proyecto.
- Cuánto tiempo se puede retrasar una actividad sin retrasar todo el proyecto.

Estas preguntas se pueden resolver utilizando el método de la ruta crítica, el cual permite determinar si una actividad del proyecto es crítica, es decir, si una demora en su comienzo causará una demora en la fecha de terminación del proyecto. Si la actividad no es crítica, tendrá un tiempo de holgura, es decir, se puede demorar.

La ruta crítica es una cadena de actividades críticas que conecta al nodo inicial con el nodo final en la red que representa al proyecto.

El método para calcular la ruta crítica de un proyecto consta de los siguientes pasos :

- 1.- Conocer el proyecto al que se aplicará el método de la ruta crítica
- 2.- Listar las actividades de este proyecto.
- 3.- Construir la matriz de secuencias.
- 4.- Construir la red de actividades.
- 5.- Numerar los nodos de la red.
- 6.- Determinar la duración de las actividades.
- 7.- Calcular el tiempo de inicio más próximo.
- 8.- Calcular el tiempo de terminación más lejano.
- 9.- Calcular la holgura total.
- 10.- Calcular la ruta crítica.

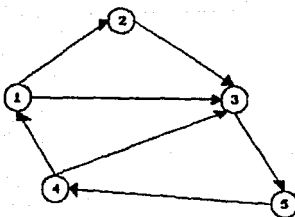
La matriz de secuencias es una tabla en la que se indica :

- Qué actividades deben seguir inmediatamente a otra actividad.
- Qué actividades deben terminarse antes de que otra actividad pueda comenzar.
- Qué actividades deben efectuarse simultáneamente con otra actividad.

ACTIVIDAD	SECUENCIA
Inicio	a
a	b
b	c
c	j, n, d
d	e, k
e	f
f	g, h
g	i
h	final
i	e, k
j	i
k	m
l	final
m	f
n	

Red de
Actividades.

Una red es un conjunto de puntos llamados nodos, que están unidos entre sí por líneas denominadas arcos. Una red se puede describir numerando los nodos y los arcos que la constituyen. Así, los nodos se enumeran o se indican con letras del alfabeto y el arco se denota por los nodos o puntos que conecta. Por ejemplo, los cinco nodos de la red que se muestra están numerados del 1 al 5 y tiene sus correspondientes arcos.



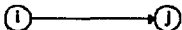
NODOS DEL SISTEMA.

ESTA TERCERA NO DEBE
SALAR DE LA BIBLIOTECA

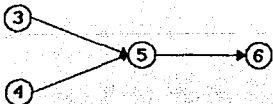
Toda la red de un proyecto debe tener un solo nodo inicial y un solo nodo final.

Todo proyecto se puede representar como una red, en donde los arcos dirigidos serán las actividades y los nodos indican el inicio y el fin de cada actividad, de manera que se puede hacer referencia a una actividad mencionando su nodo inicial y su nodo final.

La representación típica de una actividad (i,j) es :



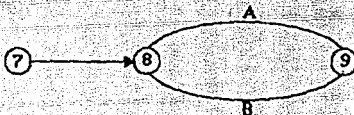
que indica que se inicia en el nodo i y termina en el nodo j. La red también muestra la secuencia de las actividades; es decir, el orden en que se deben realizar. Por ejemplo, la figura :



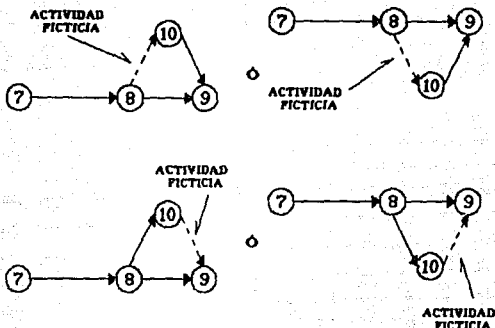
indica que las actividades (3,5) y (4,5) se deben terminar antes de que se pueda iniciar la actividad (5,6).

Las reglas para construir la red que representa el proyecto son :

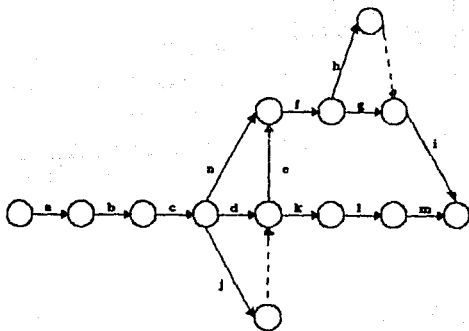
- Cada actividad está representada por uno y solamente un arco de la red. No se puede representar una misma actividad dos veces en la red.
- Dos actividades diferentes no pueden identificarse por el mismo nodo inicial y el mismo nodo final. Esta situación puede surgir cuando dos actividades deben realizarse simultáneamente.



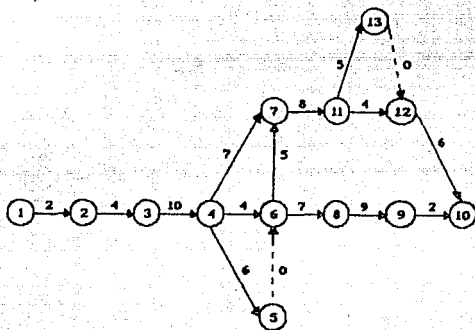
Por ejemplo, las actividades A y B pueden realizarse al terminar la actividad (7,8), simultáneamente. Ambas terminan en el nodo 9. Para evitar confusiones, se introduce una actividad ficticia al principio o al final de alguna de las actividades. Esta actividad ficticia se introduce para la construcción de la red y no tiene ningún significado, no consume ningún recurso del proyecto. La representación sería :



Se puede observar que en todos los casos se conserva la secuencia l6gica de las actividades



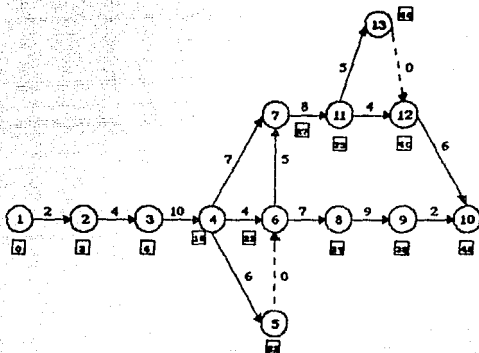
La duración de las actividades se obtiene de conocer el proyecto original y se denota como D_{ij} , y es un número que se indica sobre el arco que une dos nodos, por ejemplo :



A partir del tiempo de comienzo más próximo comienzan los cálculos de la ruta crítica, los cuales incluyen dos fases. La primera se conoce como "paso hacia adelante", donde los cálculos comienzan desde el nodo inicial y se prosigue hasta el nodo final. En cada nodo se calcula un número que representa el tiempo de comienzo más próximo de las actividades. Se indica con un cuadro (o). Si se denota este tiempo como CP_i y se inicia con $CP_1 = 0$, la forma para obtener los demás tiempos es:

$$CP_j = \max \{ CP_i + D_{ij} \}$$

La duración total del proyecto es CP_N , si N es el nodo final de la red:



$$CP_2 = \max \{ CP_1 + D_{12} \} = \max \{ 0 + 2 \} = \max \{ 2 \} = 2$$

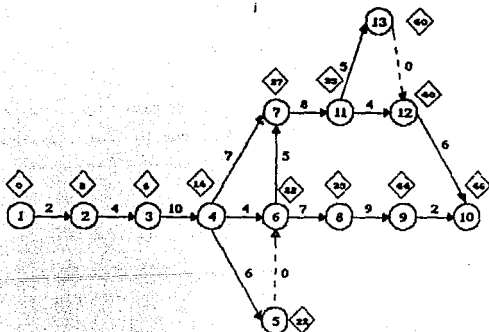
$$CP_6 = \max \{ CP_4 + D_{46}, CP_5 + D_{56} \} = \max \{ 16 + 4, 22 + 0 \} = \max \{ 20, 22 \} = 22$$

$$CP_7 = \max \{ CP_4 + D_{47}, CP_6 + D_{67} \} = \max \{ 16 + 7, 22 + 5 \} = \max \{ 23, 27 \} = 27$$

La segunda fase es llamada "paso hacia atrás" y es donde se calcula el tiempo de terminación más lejano. Los cálculos comienzan desde el nodo final y se mueve hacia el nodo inicial. En cada nodo se calcula un número, que es el tiempo de terminación más lejano de una actividad, y se indica en un rombo (∇). Si el nodo final de la red es el nodo N y el tiempo de terminación más lejano se denota como TN_j , los cálculos comienzan con $TLN = CPN$.

La fórmula para calcular los demás valores es:

$$TL_i = \min_j | TL_j - D_{ij} |$$



Por ejemplo, los cálculos se inician con :

$$TL_{10} = CP_{10} = 46$$

$$TL_{12} = \min_j | TL_{10} - D_{12} |, 10 | = \min | 46 - 6 | = 40$$

$$TL_{13} = \min_j | TL_{12} - D_{13} |, 12 | = \min | 40 - 0 | = 40$$

Antes de calcular la holgura total se deben definir dos nuevos tiempos: el tiempo de inicio más lejano CL_i y el tiempo de terminación más temprano TP_i , los que se calculan, para una actividad (i,j) , con las siguientes fórmulas :

$$CL_i = TL_j - D_{ij}$$

$$TP_i = CP_i - D_{ij}$$

Existen dos tipos de holgura: la holgura total (HT_{ij}) y la holgura libre (HL_{ij}). La holgura total HT_{ij} para la actividad (i,j) es la diferencia entre el máximo tiempo disponible para realizar la actividad $(TL_j - CP_i)$ y su duración, es decir :

$$HT_{ij} = TL_j - CP_i - D_{ij}$$

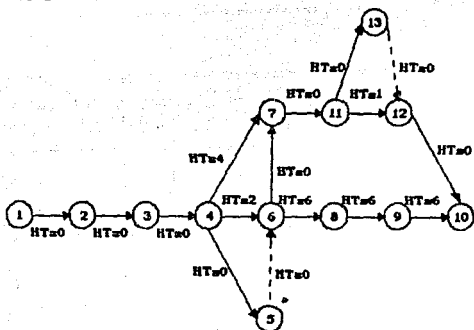
Toda aquella actividad que tenga holgura total igual a cero se dice que es una actividad crítica y queda, por lo tanto, dentro de la ruta crítica.

La holgura libre HL_{ij} se define como :

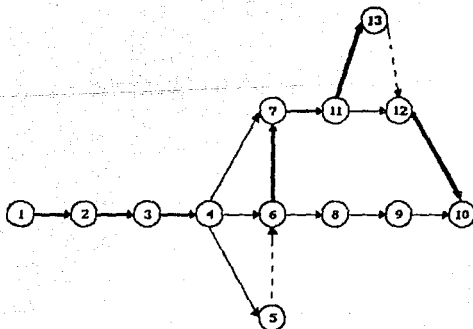
$$HL_{ij} = CP_j - CP_i - D_{ij}$$

Es decir, es el exceso de tiempo disponible menos la duración de la actividad.

Si la holgura total es cero, la holgura libre también es cero (lo inverso no es cierto).



Todas las actividades con holgura total igual a cero se encuentran dentro de la ruta crítica.



La siguiente tabla resume todos los cálculos necesarios para determinar la ruta crítica :

ACTIVIDAD (I, J)	DURACION (D _{IJ})	MAS PROMINIO		MAS LEJANO		ELOCURA	
		COMIENZO	TERM.	COMIENZO	TERM.	TOTAL	LIBRE
		CP _I <input type="checkbox"/>	TP _{IJ}	CL _I	TL _{IJ} <input checked="" type="checkbox"/>	HT _{IJ}	HL _{IJ}

TABLA PARA DETERMINAR LA RUTA CRITICA

II.2 CICLO DE VIDA.

Conforme los sistemas de cómputo se multiplican, se hacen más complejos y penetran con mayor profundidad en la sociedad moderna, se evidencia la necesidad de contar con enfoques sistemáticos para el desarrollo del software, así como para su mantenimiento. La ingeniería del software es el campo de estudio relacionado con esta nueva tecnología.

La ingeniería de software incluye la aplicación práctica del conocimiento científico en el diseño y construcción de programas para computadoras y la documentación asociada requerida para desarrollarlos, operarlos y mantenerlos.

PLANEACION DEL PROCESO DE DESARROLLO.

La planeación del proceso de desarrollo de un producto de programación comprende varias consideraciones importantes. La primera es definir un modelo para el ciclo de vida del producto. Este ciclo incluye todas las actividades requeridas para definirlo, desarrollarlo, probarlo, entregarlo, operarlo y mantenerlo. Existen diferentes modelos para el ciclo de vida pero ninguno es apropiado para todos los productos; sino que se debe optar por aquél que se ajuste a las necesidades actuales. Estos Modelos son:

Modelo de fases.

Modelo de costos.

Modelo de prototipos.

Modelo de versiones sucesivas.

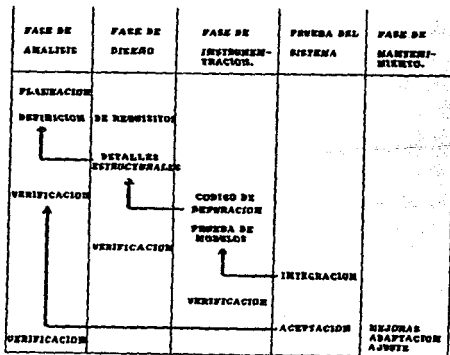
II.2.1 MODELO DE LAS FASES DEL CICLO DE VIDA.

El modelo de fases divide el ciclo de vida del producto de programación en una serie de actividades sucesivas; cada fase requiere información de entrada, procesos y resultados, todos ellos bien definidos.

Se considera el modelo de fases compuesto por las siguientes: análisis, diseño, implantación, pruebas y mantenimiento.

El análisis consta de dos subfases: planeación y definición de requisitos. Las actividades principales durante la planeación se resumen en el siguiente cuadro e incluyen la comprensión del problema del cliente, estudio de factibilidad, desarrollo de la estrategia de solución recomendada, determinación de los criterios de aceptación y planeación del proceso de desarrollo.

MODELO DE FASES PARA EL CICLO DE VIDA.



Los resultados de la planeación son la "Definición del Sistema" y el "Plan del Proyecto". La Definición, por lo regular, se expresa en español y puede contener cuadros, figuras, gráficas, etc.

El Plan del Proyecto contiene el modelo del ciclo de vida que se utilizará, la estructura organizacional del proyecto, la programación preliminar del desarrollo, estimados preliminares de costos y recursos, así como de personal, herramientas y técnicas que se emplearán y estándares que se seguirán.

Durante la fase de planeación, los estimados de costos y la programación del trabajo serán preliminares, puesto que, usualmente, no es posible realizar estimaciones precisas sin haber realizado algo del diseño.

La definición de requisitos se refiere a la identificación de las funciones básicas del componente de programación. Se pone atención en las funciones y restricciones bajo las cuales se deben de desarrollar. La decisión de cómo se instrumentará la programación se retrasa hasta la fase de diseño. El documento resultante de la definición de requisitos es una especificación que describe el ambiente de procesamiento, las funciones requeridas de los programas, restricciones de configuración sobre los programas (tamaño, velocidad, configuración de equipo), manejo de excepciones, cambios probables y modificaciones factibles, además de los criterios de aceptación del producto de programación.

El diseño se refiere a la identificación de los componentes de la programación (funciones, flujo de datos y almacenamiento), especificando las relaciones entre ellos, la estructura de la programación, y manteniendo un registro de las decisiones, proporcionando un documento base para la instrumentación. El diseño se divide en Estructural y Detallado.

El diseño estructural comprende la identificación de los componentes de la programación, su desacoplamiento y descomposición en módulos de procesamiento y estructuras de datos conceptuales, y la especificación de las interconexiones entre componentes. El diseño detallado se refiere a detalles de cómo empacar módulos de procesamiento y cómo instrumentar los algoritmos, las estructuras de datos y sus interconexiones.

Este diseño se relaciona con la adaptación del código existente, modificación de algoritmos estándar, invención de nuevos algoritmos, diseño de representaciones de datos e integración del producto final.

La fase de instrumentación en el desarrollo del producto incluye la traducción de las especificaciones del diseño en código fuente, así como su depuración, documentación y pruebas.

Los errores descubiertos durante la fase de instrumentación pueden ser errores en las interfaces de datos entre rutinas, errores lógicos en los algoritmos, errores en las estructuras de datos y de la falta de consideración de casos de procesamiento. Además, el código fuente puede contener: errores de requisitos, que indican alguna omisión de las necesidades del usuario en el documento de requisitos;

errores de diseño, que reflejan una mala traducción de requisitos en especificaciones y, por último, errores de instrumentación debido a una mala traducción. Una de las metas principales del modelo de fases es la eliminación de errores de requisitos y diseño antes de iniciada la instrumentación ya que es muy caro eliminar errores del análisis y el diseño del código fuente durante la instrumentación y las pruebas.

Las pruebas del sistema comprenden dos tipos de actividades: pruebas de integración y pruebas de aceptación. El desarrollo de una estrategia para integrar los componentes de un sistema de programación en una unidad funcional requiere una planeación cuidadosa de modo que se disponga de los módulos cuando éstos se necesiten. Las pruebas de aceptación se relacionan con la planeación y ejecución de varios tipos de pruebas para demostrar que el sistema de programación instrumentado satisface las necesidades establecidas en el documento de requisitos.

Una vez aceptado por el cliente, el sistema de programación se entrega para operación y se inicia la fase de mantenimiento. Las actividades de mantenimiento incluyen mejoras de las capacidades, adaptaciones a nuevos ambientes de procesamiento y corrección de fallas del sistema.

II.2.2 MODELO DEL COSTO DE UN PROYECTO.

Otro punto de vista para el ciclo de vida de desarrollo de un producto de programación es la consideración del costo de la realización de las distintas actividades del proyecto. El costo de un proyecto es la suma de los costos incurridos en cada fase, y éstos, a

su vez, incluyen los costos de la realización de los procesos y preparación de los documentos de esa fase más los costos de verificación de la consistencia de estos productos con los de las fases previas.

II.2.3 MODELO DE PROTOTIPO PARA EL CICLO DE VIDA.

Este modelo subraya las fuentes de requisitos para el producto. Un prototipo es una representación o modelo del producto de programación que, a diferencia de un modelo de simulación, incorpora componentes del producto real.

Por lo general, un prototipo tiene un funcionamiento limitado en cuanto a capacidades, confiabilidad y eficiencia. Sin embargo hay varias razones para desarrollar un prototipo. Una de ellas es ilustrar los formatos de los datos de entrada, mensajes, informes y diálogos al cliente y tener un mejor entendimiento de las necesidades de él. Otra razón es explorar aspectos técnicos del producto propuesto.

II.2.4 VERSIONES SUCESIVAS.

El método de versiones sucesivas es una extensión del modelo de prototipos en el que se refina un esqueleto inicial del producto, obteniendo así cada vez más capacidades. En dicho método, cada versión es un sistema funcional capaz de realizar trabajo útil.

II.2.6 DISEÑO DE SOFTWARE.

El proceso de un diseño incluye concebir y planear algo en la mente, así como hacer un dibujo, modelo o croquis . La abstracción es la herramienta que permite trabajar con los conceptos independientemente de las instancias particulares de éstos. Durante la definición de los requerimientos y el diseño, la abstracción permite la separación de los aspectos conceptuales de un sistema de los que será más tarde instrumentado.

ESTRUCTURA.

La estructura es una característica fundamental de los productos de programación. El uso de una estructuración permite que un sistema grande sea definido en términos de unidades más pequeñas y manejables con una clara definición de las relaciones entre las diferentes partes del sistema.

MODULARIDAD.

Los sistemas modulares consisten en unidades bien definidas y manejables con las interfaces claramente definidas entre los diversos módulos. La modularidad mejora el diseño, que, a su vez, facilita la instrumentación, la depuración, las pruebas, la documentación y el mantenimiento de un producto de programación.

VERIFICACION.

La verificación es un concepto fundamental en el diseño de la programación, ya que el diseño es el puente entre los requerimientos del cliente y la instrumentación que satisface esos requisitos. Un diseño es verificable si puede demostrarse que generará el producto que satisface los requerimientos del cliente. Esto se desarrolla comúnmente en dos pasos:

- Verificación que la definición de los requisitos de programación satisface las necesidades del usuario.
- Verificación que el diseño satisface la definición de los requisitos.

ESTETICA.

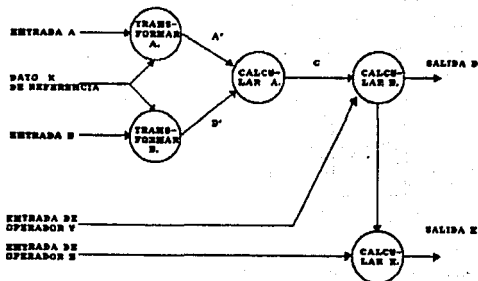
Tanto en las artes como en la ingeniería, las consideraciones estéticas son fundamentales para el diseño. La simplicidad, elegancia y claridad de un propósito distinguen a los productos de alta calidad de los mediocres.

NOTACIONES PARA EL DISERO.

En el diseño de la programación, como en las matemáticas, los esquemas de representación utilizados son de fundamental importancia. Una buena notación puede clarificar las interrelaciones y las interacciones de interés, mientras que una notación pobre puede complicar e interferir con la buena práctica de la programación.

DIAGRAMAS DE FLUJOS DE DATOS.

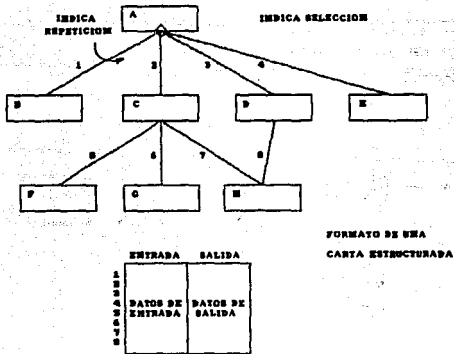
Los diagramas de flujos de datos o burbujas son gráficas dirigidas en donde los nodos especifican las actividades de proceso y los arcos la transferencia de datos entre nodos de proceso. Como cualquier otro diagrama de flujo, éstos pueden ser usados en cualquier nivel de abstracción.



CARTAS DE ESTRUCTURA.

Las cartas de estructura son utilizadas durante el diseño para documentar la estructura jerárquica, los parámetros y las interconexiones dentro de un sistema. Una carta de estructura difiere de un diagrama de flujo clásico en dos aspectos: no tiene diagramas

de decisión y el orden secuencial de las tareas inherentes en un diagrama de flujo pueden ser eliminadas de una tabla de estructura.



SEUDOCÓDIGO.

El seudocódigo, al igual que los diagramas de flujo, puede ser utilizado en cualquier nivel de abstracción. Con el uso del seudocódigo el diseñador describe las características del sistema usando frases cortas y concisas en español, las cuales se encuentran estructuradas por medio del uso de palabras clave como : si-entonces-si-no, mientras-repetir y fin. Con estas palabras y el uso del sangrado se puede describir el flujo del control del programa, mientras que cada frase en español describe las acciones a ejecutar.

INICIAR tablas y contadores; **ABRIR** archivos
LEER el primer registro del texto
MIENTRAS haya más registros de texto **REPETIR**

FINMIENTRAS cuando todos los registros de texto han sido
procesados

IMPRIMIR la tabla de palabras y el contador de palabras procesadas

CERRAR archivos

TERMINAR el programa

II.3 TEORIA GENERAL DE BASES DE DATOS.

Un sistema manejador de bases de datos (DBMS) está constituido por un conjunto de datos interrelacionados y un conjunto de programas que permiten el acceso a tales datos. La colección de datos es conocida como una base de datos. La base de datos contiene información acerca de algún negocio en particular. El objetivo principal de un DBMS es proporcionar un medio conveniente y eficiente que permita la recuperación y el almacenamiento de información en la base de datos.

Los sistemas de bases de datos se diseñan para poder manejar grandes cantidades de información. El manejo de los datos involucra tanto la definición de las estructuras para el almacenamiento de la información como los mecanismos para la manipulación de la información. Además, el sistema debe proporcionar seguridad para la información almacenada en la base de datos, ya sea que el sistema se dañe o se intente un acceso no autorizado. Asimismo, si los datos serán compartidos entre varios usuarios el sistema debe evitar posibles resultados erróneos.

Debido a la importancia de la información en la mayoría de las empresas, la base de datos es un recurso muy valioso. Esto ha llevado al desarrollo de una gran cantidad de conceptos y técnicas para hacer posible un manejo eficiente de los datos.

OBJETIVO DE UN SISTEMA DE BASE DE DATOS.

Considérese parte de un banco que maneja información de sus clientes, así como de sus cuentas. La información personal de los clientes y la de las cantidades de éstos en el banco se almacenan en sistemas de archivos permanentes. Además de estos archivos, el sistema tiene un número de programas de aplicación que permiten manipular los archivos. Algunos de los programas incluidos son :

Un programa para agregar una nueva cantidad.

Un programa para encontrar el balance de una cuenta.

Un programa para generar avisos mensuales.

Estos programas de aplicación son escritos por programadores con el fin de satisfacer las necesidades del banco.

Nuevos programas de aplicación se suman a los ya existentes conforme las necesidades aumentan. Ya que estos programas y archivos se van creando durante un período grande de tiempo (presumiblemente por diferentes programadores), es común que los archivos tengan diferente formato y que los programas sean escritos en diferentes lenguajes de programación.

El medio ambiente descrito anteriormente es conocido como un Sistema de Procesamiento de Archivos, el cual es soportado por un sistema operativo convencional. Sin embargo, este esquema tiene un número de desventajas :

• Redundancia e inconsistencia de datos. Ya que los archivos y los programas son creados por diferentes personas, una misma pieza de información puede estar duplicada en diferentes lugares (archivos). Esta redundancia de la información hace necesario el uso de mayores recursos de almacenamiento y que el acceso a la información sea más costosa y, a la vez, se corre el riesgo de tener inconsistencia en los datos.

• Aislamiento de la información. Como los datos están almacenados en diferentes archivos, y los archivos pueden tener diferente formato, se dificulta el escribir nuevos programas para recuperar los datos.

• Usuarios múltiples. Con el fin de mejorar la eficiencia del sistema y obtener un mejor tiempo de respuesta, muchos sistemas permiten que múltiples usuarios puedan acceder y actualizar datos simultáneamente. En un medio ambiente como el descrito, la concurrencia de múltiples actualizaciones puede resultar en inconsistencia de datos.

• Problemas de seguridad. No todo usuario del sistema debe ser capaz de acceder toda la información, pero debido a que los programas fueron sumados al sistema conforme se necesitaban es muy difícil implementar un mecanismo de seguridad.

• Problemas de integridad. Los valores de los datos almacenados en la base de datos deben satisfacer ciertas restricciones de integridad. Por ejemplo, el balance de una cuenta bancaria nunca

debe ser menor que una cantidad especificada previamente (por ejemplo \$25000). Estas restricciones deben ser implementadas de alguna manera en el sistema. Sin embargo, cuando se requieren nuevas restricciones es muy difícil cambiar los programas para implementarlas.

Estas dificultades, entre otras, son las que han conducido al desarrollo de los sistemas manejadores de bases de datos.

II.3.1 ABSTRACCION DE DATOS.

Un sistema manejador de bases de datos es un conjunto de archivos interrelacionados y un conjunto de programas que permiten que varios usuarios accedan y modifiquen estos archivos. Un propósito fundamental de un sistema de bases de datos es proporcionar a los usuarios una vista abstracta de los datos. Esto es, el sistema oculta ciertos detalles de la forma en que la información es almacenada y actualizada. Sin embargo, con el propósito de que el sistema sea útil, los datos deben ser recuperados eficientemente.

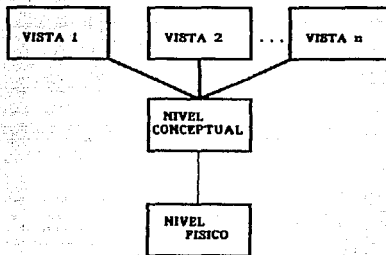
El manejo de la eficiencia lleva al diseño de estructuras complejas para la representación de los datos en la base de datos. No obstante, ya que los sistemas de bases de datos son frecuentemente usados por personas sin experiencia en computación, esta complejidad debe ocultarse a los usuarios. Esto se puede lograr definiendo varios niveles de abstracción desde los cuales puede ser vista la base de datos.

• Nivel físico. Este es el más bajo nivel de abstracción en el que se describe cómo están almacenados los datos. En este nivel se describen a detalle las estructuras de datos más simples.

• Nivel conceptual. Este es el siguiente nivel de abstracción en el cual se describe qué datos están almacenados y las relaciones que existen entre los mismos. Este nivel describe a la base de datos completa en términos de un número pequeño de estructuras relativamente simples. Aunque la implementación de estas estructuras puede involucrar estructuras del nivel físico, el usuario del nivel físico no necesita conocer mucho de ellas. Este nivel es usado por los administradores de la base de datos, ya que son ellos quienes deciden qué información debe ser guardada en la base de datos.

• Nivel visual. Es el más alto nivel de abstracción en el que se describe sólo parte de la base de datos. No obstante el uso de estructuras simples en este nivel, aún persiste cierta complejidad debida al tamaño de la base de datos. En tal caso, los usuarios no manejan la totalidad de la información. Para facilitar la interacción del usuario con la base de datos, el nivel de abstracción proporciona varias vistas para la misma base de datos.

La interrelación entre estos tres niveles de abstracción se ilustra en la siguiente figura:



II. 3.2 MODELOS DE DATOS.

Con el propósito de definir la estructura de una base de datos es necesario definir el concepto de un Modelo de Datos. Un modelo de datos es una colección de herramientas conceptuales para describir los datos, las relaciones entre los datos y las restricciones de los datos. Existen diferentes modelos que han sido propuestos y que se clasifican en tres diferentes grupos : Modelos lógicos orientados a objetos, Modelos lógicos orientados a registros y Modelos de datos físicos.

MODELOS LOGICOS ORIENTADOS A OBJETOS.

Estos modelos se usan en la descripción de datos en los niveles conceptual y visual. Se caracterizan por el hecho de que proporcionan bastante flexibilidad y capacidad de estructuración y permiten la especificación explícita de restricciones de datos. Por lo menos existen treinta diferentes modelos. Entre los más conocidos están :

- Modelo entidad-relación.
- Modelo binario.
- Modelo semántico.
- Modelo infológico.

De éstos, el modelo entidad-relación es el más representativo.

El modelo entidad-relación (E-R) se basa en una percepción del mundo real, es decir, en la percepción de un conjunto de objetos básicos llamados entidades y un conjunto de relaciones entre estos objetos. Una entidad es un objeto que existe y que puede distinguirse de otros objetos. Esta distinción se realiza al asociar cada entidad con un conjunto de atributos que describen al objeto. Por ejemplo, los atributos número y balance describen una cuenta particular en un banco. Una relación es una asociación entre varias entidades. El conjunto de todas las entidades del mismo tipo y el conjunto de relaciones del mismo tipo son llamados conjunto de entidades y conjunto de relaciones, respectivamente. Además de las entidades y relaciones, el modelo E-R representa ciertas restricciones que el contenido de la base de datos debe cumplir. Una restricción muy importante es el mapeo de cardinalidad, la cual expresa el número de entidades con las que otra entidad puede asociarse por medio de un conjunto de relaciones. La estructura lógica de una base de datos puede expresarse gráficamente por medio de un diagrama E-R y que básicamente contiene los siguientes componentes :

- Rectángulos, que representan conjuntos de entidades.
- Elipses, que representan a los atributos.

• Rombos, que representan las relaciones entre conjuntos de entidades.

• Líneas, que asocian los atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones.

Cada componente se etiqueta con su nombre correspondiente. Para ilustrar lo anterior, considérese parte de un sistema de base de datos de un banco, el cual contiene clientes y las cuentas de éstos.

El diagrama E-R correspondiente a este esquema es el siguiente :

Diagrama Entidad Relacion (E - R)

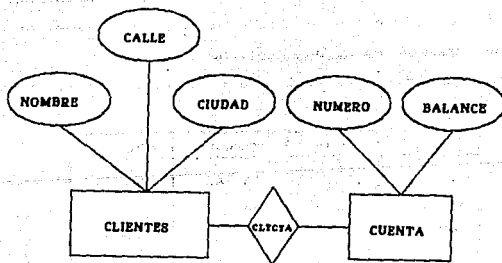


DIAGRAMA E-R.

MODELOS LOGICOS ORIENTADOS A REGISTROS.

Estos modelos se usan en la descripción de datos en los niveles conceptual y visual. En contraste con los modelos orientados a objetos, estos modelos se usan tanto para la especificación de la estructura lógica de la base de datos como para la descripción de la implementación. Sin embargo, estos no proporcionan facilidades para la especificación de restricciones explícitamente. Los tres modelos más ampliamente aceptados son : Modelo relacional, Modelo de red y Modelo jerárquico.

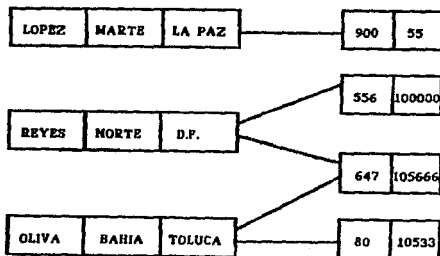
Modelo relacional. Los datos y las relaciones entre estos se representan por medio de un conjunto de tablas. Cada tabla contiene un número de columnas con un nombre único. Para ilustrar esto, considérese una base de datos que contiene información de clientes y sus respectivas cuentas bancarias. Esto se puede ver en las siguientes tablas.

NOMBRE	CALLE	CIUDAD	NUMERO
LOPEZ	MARTE	LA PAZ	900
REYES	NORTE	D.F.	556
REYES	NORTE	D.F.	647
OLIVA	BAHIA	TOLUCA	80
OLIVA	BAHIA	TOLUCA	647

NUMERO	BALANCE
900	55
556	100000
647	105366
80	10533

Como se observa, el cliente Oliva vive en Toluca; posee dos cuentas bancarias : la 647 con un balance de \$105366 y la 80 con un balance de \$10533. Notese que los clientes Oliva y Reyes comparten la cuenta 647.

Modelo de red. Los datos en el modelo de red se representan por colecciones de registros y las relaciones entre los datos se representan por ligas, las que pueden ser conceptualizadas como apuntadores. Los registros de la base de datos se organizan en forma de conjuntos de registros arbitrarios. La información representada en el modelo relacional se puede especificar en el modelo de red de la siguiente manera.



MODELO DE RED.1

INSTANCIAS Y ESQUEMAS.

Las bases de datos cambian a través del tiempo conforme la información va siendo almacenada o borrada de ésta. El conjunto de información almacenada en la base de datos en un instante determinado de tiempo es llamado una Instancia de la base de datos. El diseño completo de la base de datos es llamado Esquema de la Base de Datos.

El concepto de "esquema" de base de datos corresponde con la definición de un tipo de registro en lenguajes de programación. Una variable de un tipo dado tiene un valor particular en algún instante de tiempo. El concepto de valor de una variable en lenguajes de programación corresponde al concepto de instancia de un esquema de base de datos.

Existen varios esquemas en bases de datos, y éstos se dividen de acuerdo a los niveles de abstracción mencionados anteriormente. En el nivel inferior se tiene el esquema físico; en el nivel intermedio se tiene el esquema conceptual; mientras que en el nivel superior se tiene el subesquema. En general, los sistemas de bases de datos soportan un esquema físico, un esquema conceptual y varios subesquemas.

INDEPENDENCIA DE DATOS.

Como se especificó anteriormente, hay tres niveles de abstracción desde los cuales puede ser vista una base de datos. La capacidad para modificar la definición de un esquema en un nivel sin afectar la definición de un esquema en el nivel siguiente es llamado Independencia de datos.

Existen dos niveles de independencia de datos :

- Independencia de datos físicos, es la capacidad para modificar el esquema físico sin provocar que los programas sean reescritos. Las modificaciones en el nivel físico se requieren ocasionalmente con el propósito de mejorar la eficiencia.

- Independencia de datos lógicos, es la capacidad para modificar el esquema conceptual sin provocar que los programas sean reescritos. Las modificaciones en el nivel conceptual se requieren siempre que la estructura lógica de la base de datos es alterada.

La independencia de datos lógicos es más difícil de obtener debido a que los programas de aplicación son más dependientes de la estructura lógica de los datos que accesan.

El concepto de independencia es similar en muchos aspectos al concepto de Tipos de datos abstractos usado en lenguajes de programación modernos. En ambos casos se oculta a los usuarios los detalles de la implementación. Esto permite que los usuarios se concentren en la estructura general y no en los detalles de implementación en un nivel inferior.

MODELO ENTIDAD-RELACION.

El modelo de datos Entidad-Relación (E-R) se basa en una percepción del mundo real, el que está formado por un conjunto de objetos básicos llamados entidades y relaciones entre estos objetos. Este modelo fue desarrollado con el propósito de facilitar el diseño de bases de datos al permitir la especificación de un esquema que representa la totalidad de la estructura lógica de la base de datos.

ENTIDADES Y CONJUNTOS DE ENTIDADES.

Una entidad es un objeto que existe y que puede distinguirse de otros objetos. Por ejemplo, Juan Ruiz con el número de registro B90123456 es una entidad, ya que identifica a una persona en particular dentro del universo. Una entidad puede ser algo concreto, tal como una persona o un libro, o bien algo abstracto, como la fecha de cumpleaños o algún otro concepto.

Un conjunto de entidades es un conjunto de entidades del mismo tipo. El conjunto de todas las personas que tienen cuenta en un banco, por ejemplo, puede ser definido como el conjunto de entidades clientes. Es posible definir un conjunto de entidades de todos los empleados del banco (empleados) y un conjunto de entidades de los clientes del banco (clientes); la entidad persona puede ser una entidad de empleados, una entidad de clientes, de ambos o de ninguno.

Una entidad se representa por un conjunto de atributos. Así, los posibles atributos del conjunto de entidades clientes son : nombre, seguro-social, calle y ciudad. Para cada atributo existe un conjunto de valores permitidos, llamado dominio del atributo. El dominio del atributo nombre puede ser todos los nombres de personas.

Formalmente, un atributo es una función que mapea un conjunto de entidades dentro de un dominio. En consecuencia, toda entidad se describe por un conjunto de pares (atributo,valor), un par para cada atributo del conjunto de entidades. Por ejemplo, una entidad de clientes se describe por el conjunto{(nombre,Juan), (seguro-social,89012), (calle,Sur), (ciudad,D.F.)}, lo que significa que la entidad describe a la persona de nombre Juan con número de seguro social 89012 y que reside en la calle Sur en el D.F.

El concepto de conjunto de entidades es análogo a la noción de definición de tipos en lenguajes de programación, mientras que el concepto de entidad en el modelo E-R corresponde al concepto de variable.

Una base de datos incluye una colección de conjuntos de entidades, cada uno de los cuales contiene un número de entidades del mismo tipo.

A continuación se muestra parte de una base de datos de un banco que contiene dos conjuntos de entidades : clientes y cuentas.

RAUL	654-32	NORTE	LINDAVIS
JUAN	890-12	SUR	PERISUR
MARIO	456-78	NORTE	VILLA M.
BERGIO	369-12	SUR	TLALPAM
JULIO	246-80	ORIENTE	ZARAGOZ
ROBERTO	121-21	ESTE	CUITLAH.
JORGE	135-79	SUR	UNIVERE.

CLIENTE

259	1000
630	2000
401	1500
700	1500
199	500
167	900
115	1200
103	1300
118	2000
225	2500
210	2220

CUENTA

RELACIONES Y CONJUNTOS DE RELACIONES.

Una relación es una asociación entre varias entidades. Por ejemplo, se puede definir una relación que asocia el cliente 'Juan' con la cuenta 401. Esto especifica que Juan es un cliente con número de cuenta bancaria 401.

Un conjunto de relaciones es un conjunto de relaciones del mismo tipo. Formalmente es una función matemática en $n \geq 2$ conjuntos de entidades. Si E_1, E_2, \dots, E_n son conjuntos de entidades, entonces un conjunto de entidades R es un subconjunto de

$$\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$$

donde (e_1, e_2, \dots, e_n) es una relación.

Para ilustrar lo anterior, consideremos los dos conjuntos de entidades mostrados en la figura anterior. Definamos la relación CiteCta para denotar la asociación entre clientes y las cuentas que éstos tienen en el banco. Esta asociación se ilustra en la siguiente figura.

CLIENTE	RAUL	654-32	NORTE	LINDAVIS	259	1000	CUENTA
					630	2000	
	JUAN	890-12	SUR	PERISUR	401	1500	
	MARIO	456-78	NORTE	VILLA N.	700	1500	
					199	500	
	SERGIO	359-12	SUR	TLALPAN	167	900	
	JULIO	246-80	ORIENTE	ZARAGOZ	115	1200	
					103	1300	
ROBERTO	121-21	ESTE	CUITLAH.	118	2000		
JORGE	135-79	SUR	UNIVERS.	225	2500		
				210	2220		

La relación CiteCta es un ejemplo de un conjunto de relaciones binario, esto es, una relación que involucra dos conjuntos de entidades. La mayoría de los conjuntos de relaciones en un sistema de base de datos son binarios. Ocasionalmente, sin embargo, hay conjuntos de relaciones que involucran más de dos conjuntos de entidades. Como un ejemplo, consideremos la relación ternaria (Juan, 401, Perisur), que especifica que el cliente Juan tiene la cuenta 401 en la sucursal Perisur. Esta relación es una instancia de la relación CiteCtaSuc que involucra tres conjuntos de entidades.

La función de una entidad dentro de una relación es llamada rol. Los roles normalmente son implícitos y no hay necesidad de especificarlos.

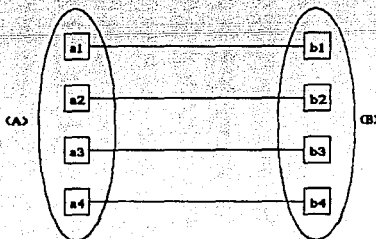
11.3.3 RESTRICCIONES DE MAPEO.

Un esquema E-R puede definir ciertas restricciones a las que está sujeta una base de datos. Una restricción importante es el mapeo de cardinalidad, la que especifica el número de entidades con las que otra entidad puede ser asociada por medio de una relación.

El mapeo de cardinalidades es muy útil en la descripción de conjuntos de relaciones binarias, aunque ocasionalmente sirve en la descripción de conjuntos de relaciones que involucran más de dos conjuntos de entidades.

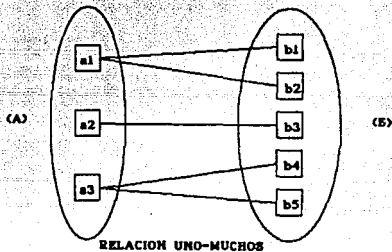
Para un conjunto de relaciones binarias R entre los conjuntos de entidades A y B , el mapeo de cardinalidad debe ser alguno de los siguientes:

Uno-uno. Una entidad en A es asociada con, a lo máximo, una entidad en B , y una entidad en B es asociada con, a lo máximo, una entidad en A .

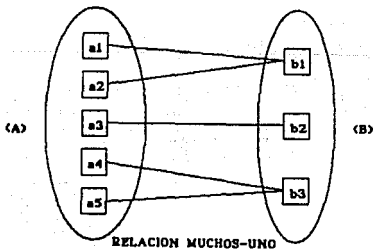


RELACION UNO-UNO

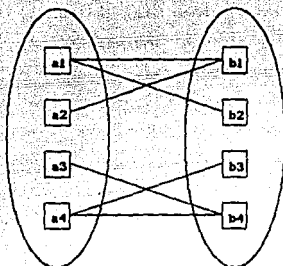
Uno-muchos. Una entidad en A es asociada con cualquier número de entidades en B. Una entidad en B, sin embargo, puede ser asociada con, a lo máximo, una entidad en A.



Muchos-uno. Una entidad en A es asociada con, a lo máximo, una entidad en B. Una entidad en B, sin embargo, puede ser asociada con cualquier número de entidades en A.



Muchos-muchos. Una entidad en A es asociada con cualquier número de entidades en B y una entidad en B puede ser asociada con cualquier número de entidades en A.



RELACION MUCHOS-MUCHOS

El mapeo de cardinalidad apropiado para un conjunto de relaciones en particular depende del mundo real que se esté modelando.

Para ilustrar lo anterior, considérese el conjunto de relaciones CiteCta. Si en un banco en particular una cuenta puede pertenecer a solo un cliente, y un cliente puede tener varias cuentas, entonces el conjunto de relaciones es uno a muchos.

Otro tipo de restricciones importantes son las dependencias de existencia. Si la existencia de una entidad x depende de la existencia de una entidad y, entonces se dice que la existencia de x depende de y. Esto significa que si y es borrada, también lo será x. En este caso se dice que y es una entidad dominante y x es una entidad subordinada.

LLAVES PRIMARIAS.

Una tarea importante en el modelado de bases de datos es especificar cómo se distinguen las entidades de las relaciones. Conceptualmente, las entidades individuales y las relaciones son distintas, pero desde una perspectiva de bases de datos la diferencia entre éstas debe ser expresada en términos de sus atributos. Para establecer esta distinción, una superllave se asigna a cada conjunto de entidades. La superllave es un conjunto de uno o más atributos, los cuales, tomados en conjunto, nos permiten identificar de forma única una entidad del conjunto de entidades. Por ejemplo, el atributo seguro-social del conjunto de entidades clientes es suficiente para distinguir una entidad de otra. De esta manera, seguro-social es una superllave. Similarmente, la combinación de los atributos nombre-cliente y seguro-social es una superllave para el conjunto de entidades clientes. El atributo nombre-cliente no es una superllave ya que varias personas pueden tener el mismo nombre. Es frecuente que se busque tener superllaves mínimas, es decir, que estén formadas por el menor número de atributos. Tales superllaves mínimas son llamadas llaves candidato. Es posible que existan diferentes conjuntos de atributos que puedan servir como llaves candidato.

El término llave primaria se usa para denotar una llave candidato que es seleccionada como medio principal para identificar entidades dentro de un conjunto de entidades.

Es posible que se presente el caso en que un conjunto de entidades no tenga los atributos suficientes para formar una llave primaria. Tales entidades son llamadas entidades débiles. Una entidad que tiene una llave primaria se llama entidad fuerte. El concepto de entidad débil y fuerte está relacionado con el concepto de dependencia de existencia mencionado anteriormente. Una entidad fuerte es, por definición, una entidad dominante, mientras que una entidad débil es una entidad subordinada.

Un conjunto de entidades débiles no tiene una llave primaria. Sin embargo, se necesita una manera de distinguir, entre todas las entidades, las que dependen de una entidad fuerte. El discriminador de una entidad débil es un conjunto de atributos que nos permite hacer esta distinción. La llave primaria de un conjunto de entidades débiles se forma con la llave primaria de la entidad fuerte de la cual depende la existencia de tal conjunto, más su discriminador.

Los conjuntos de relaciones también tienen llaves primarias. Sus llaves primarias se forman al tomar todos los atributos que conforman la llave primaria del conjunto de entidades que definen al conjunto de relaciones.

II.3.4 DIAGRAMA ENTIDAD-RELACION.

La totalidad de la estructura lógica de una base de datos puede ser expresada gráficamente por medio de un diagrama entidad-relación, el cual está formado por los siguientes componentes :

- Rectángulos, que representan conjuntos de entidades.
- Elipses, que representan atributos

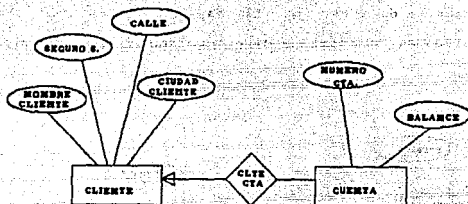
- **Rombos**, que representan las relaciones entre los conjuntos de atributos.
- **Líneas**, ligan los atributos con los conjuntos de entidades, y los conjuntos de entidades con los conjuntos de relaciones.

Cada componente se etiqueta con su nombre correspondiente. Para ilustrar esto, considérese el siguiente diagrama E-R, el cual tiene dos conjuntos de entidades : clientes y cuentas, relacionados por medio del conjunto de relaciones binarias ClteCta. Los atributos asociados con clientes son nombre-cliente, seguro-social, calle y ciudad. Los atributos asociados con cuentas son número-cuenta y balance.

El conjunto de relaciones Clte-Cta puede ser muchos-muchos, uno-muchos, muchos-uno, o uno-uno. Para diferenciar entre estos tipos, se dibuja una línea con flecha o una línea sin flecha entre el conjunto de relaciones y el conjunto de entidades en cuestión. Una línea con flecha del conjunto de relaciones Clte-Cta al conjunto de entidades cuenta, especifica que el conjunto de entidades cuenta participa en una relación uno-uno o muchos-uno con el conjunto de entidades clientes. Una línea sin flecha del conjunto de relaciones Clte-Cta al conjunto de entidades cuenta especifica que el conjunto de entidades cuentas participa en una relación muchos-muchos o muchos-uno con el conjunto de entidades clientes.

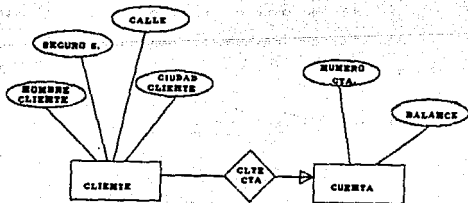
En el anterior diagrama E-R se puede ver que el conjunto de relaciones "cliente-cuenta" es muchos-muchos. Si el conjunto de relaciones fuera uno-muchos, de clientes a cuentas, entonces la liga

de "CiteCta" debería tener una flecha apuntando al conjunto de entidades "Clientes", como se muestra a continuación.



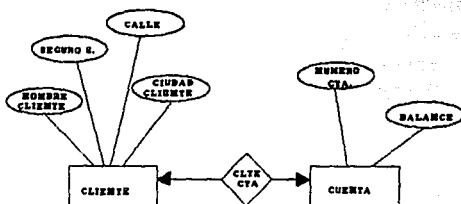
RELACION UNO - MUCHOS

Similarmente, si el conjunto de relaciones "ClteCta" fuera muchos-uno de clientes a cuentas, la liga de cliente cuenta debería tener una flecha hacia el conjunto de entidades "cuentas", como en la siguiente figura :



RELACION MUCHOS - UNO

Finalmente, si el conjunto de relaciones fuera uno-uno, la liga de "CiteCta" debería de tener dos flechas, una apuntando hacia el conjunto de entidades "cuentas" y otra apuntando hacia el conjunto de entidades "clientes", como en la siguiente figura :



RELACION UNO - UNO

En un diagrama E-R, los roles se indican con etiquetas sobre las líneas que conectan los rectángulos con los rombos. En la siguiente figura se muestra el conjunto de entidades "empleados" y el conjunto de relaciones "trabaja-para".

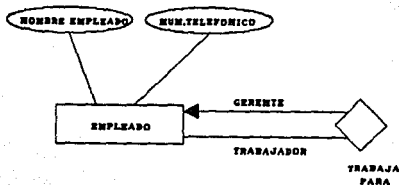


DIAGRAMA E-R CON INDICADORES DE ROLES.

Un conjunto de entidades débiles es indicado en un diagrama E-R mediante un rectángulo con doble línea. La relación que conecta a éste con el conjunto de entidades fuertes con el cual se forma su llave primaria se indica con líneas sombreadadas. En la siguiente figura se muestra el conjunto de entidades débiles "transacciones" que depende del conjunto de entidades fuertes "cuentas" por medio de la relación "log".

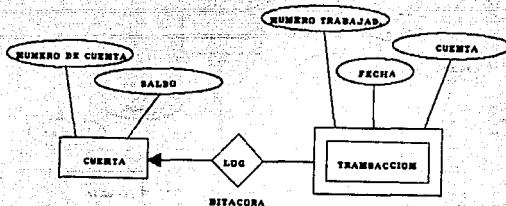


DIAGRAMA E-R CONJUNTO DE ENTIDADES DEBILES

El siguiente esquema muestra un conjunto de relaciones no binario, que contiene tres conjuntos de entidades: Clientes, Cuentas y Sucursales, relacionados por medio del conjunto de relaciones CCS. Este diagrama especifica que un cliente puede tener varias cuentas, cada una en diferente sucursal, y que una cuenta puede pertenecer a varios clientes.

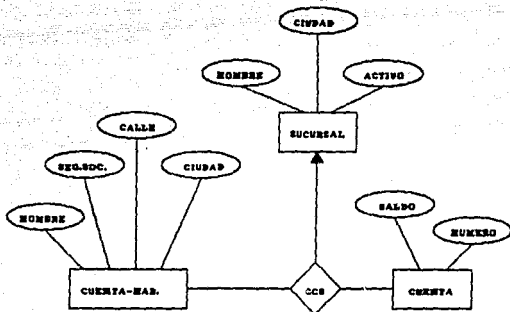


DIAGRAMA E - R CON RELACION TERNARIA.

REDUCCION DE DIAGRAMAS ENTIDAD RELACION A TABLAS.

Una base de datos conceptualizada por medio de un diagrama E-R puede representarse por medio de un conjunto de tablas. Para cada conjunto de entidades y para cada conjunto de relaciones en la base de datos existe una tabla única a la que se le asigna el nombre correspondiente del conjunto de entidades o del conjunto de relaciones, cada tabla tiene un número de columnas que, a su vez, tienen nombres únicos.

REPRESENTACION DE CONJUNTOS DE ENTIDADES FUERTES.

Sea E un conjunto de entidades fuertes con atributos descriptivos a_1, a_2, \dots, a_n . Este conjunto de entidades se representa por una tabla llamada E con n columnas, cada una de las cuales corresponde a uno de los atributos de E . Cada renglón de esta tabla corresponde a una entidad del conjunto de entidades E .

Considérese el conjunto de entidades "cuenta" del diagrama $E-R$ de la página anterior. Este conjunto de entidades tiene dos atributos: "número-cta" y "saldo". Este conjunto de entidades se representa por medio de la tabla llamada cuentas, con dos columnas como se muestra a continuación.

NUMERO CTA.	SALDO
259	1000
630	2000
401	1500
700	1500
199	500
467	900
115	1200
183	1300
118	2000

TABLA
" CUENTAS "

El renglón (259,1000) significa que el número de cuenta 259 tiene un balance de \$1000 pesos. Se puede agregar nuevas entidades a la base de datos al insertar renglones en la tabla. También se puede borrar o modificar renglones.

Sea D_1 que denota al conjunto de todos los números de cuentas y D_2 que denota al conjunto de balances. Cualquier renglón del conjunto cuentas debe ser un 2-tuplo (v_1, v_2) , donde v_1 es un número de cuenta

(esto es, v_1 está en el conjunto de D_1) y v_2 es un balance (esto es, v_2 está en el conjunto D_2). En general la tabla de cuentas solo contendrá un subconjunto del conjunto de todos los renglones posibles. Se hace referencia al conjunto de todos los posibles renglones de la tabla de cuentas como el producto cartesiano de D_1 y D_2 , lo que se denota como :

$$D_1 \times D_2$$

o bien

$$2$$

$$\times D_1$$

$$i=1$$

En general, si se tiene una tabla de n columnas, el producto cartesiano de D_1, D_2, \dots, D_n se denota como :

$$n$$

$$\times D_i$$

$$i=1$$

Como otro ejemplo, considérese el conjunto de entidades clientes con sus cuatro atributos descriptivos : "nombre-cliente", "seguro-social", "calle" y "ciudad". la tabla correspondiente es la siguiente:

NOM. CTAHAB.	NUM. SEG. SOC.	CALLE	CIUDAD CTAHAB.
OLIVA	654-32	5 MAYO	PUEBLA
HERRERA	890-12	NORTE 5	D.F.
MANZO	456-78	VALMIS	TOLUCA
PEDRAZA	369-12	RIO	CUERNAVACA

CONJUNTO DE ENTIDADES "CLIENTES"

REPRESENTACION DE CONJUNTOS DE ENTIDADES DEBILES.

Sea A un conjunto de entidades débiles con los atributos descriptivos a_1, a_2, \dots, a_r . Sea B un conjunto de entidades fuertes del cual depende A. Sea la llave primaria de B formada por los atributos b_1, b_2, \dots, b_s . El conjunto de entidades A se representa por medio de una tabla llamada A con una columna por cada atributo del conjunto

$$\{ a_1, a_2, \dots, a_r \} \cup \{ b_1, b_2, \dots, b_s \}$$

Para ilustrar esto, considere el conjunto de entidades "transacciones" del anterior diagrama E-R. Este conjunto de entidades tiene 3 atributos: "número-transacción", "fecha" y "cantidad". La llave primaria del conjunto entidades "cuentas", de la cual depende "transacciones", es "número-cuenta". De esta manera, "transacciones" se representan por medio de una tabla con 4 columnas como se muestra a continuación.

NUM. CUENTA	NUM. TRANSACCION	FECHA	IMPORTE
259	5	15 MAYO	+50
630	11	19-MAYO	+70
401	22	27-MAYO	-300
700	69	03-JUNIO	-500
199	103	09-JUNIO	-900

REPRESENTACION DE CONJUNTOS DE RELACIONES.

Sea R un conjunto de relaciones que involucra a los conjuntos de entidades E_1, E_2, \dots, E_n . Sea llave-primaria (E_i) el conjunto de atributos que forman la llave primaria del conjunto de entidades E_i . Supóngase que R no tiene atributos descriptivos. La tabla correspondiente al conjunto de relaciones R tiene los siguientes conjuntos de atributos

$$\bigcup_{i=1}^n \text{llave-primaria}(E_i)$$

En el caso en que R tiene atributos descriptivos, por ejemplo $\{a_1, a_2, \dots, a_m\}$, entonces la tabla correspondiente tiene los siguientes atributos:

$$\bigcup_{i=1}^n \text{llave-primaria}(E_i) \cup \{a_1, a_2, \dots, a_m\}$$

Para ilustrar esto, considérese el conjunto de relaciones "CiteCta" del anterior diagrama E-R. Este conjunto de relaciones involucra a los siguientes dos conjuntos de entidades :

- Clientes, con llave primaria "seguro-social"
- Cuentas, con llave primaria "número-cuenta"

Ya que el conjunto de relaciones tiene el atributo descriptivo "fecha", la tabla "CiteCta" tiene tres columnas.

II.4 MODELO RELACIONAL.

Desde una perspectiva histórica el modelo de datos relacional es relativamente reciente. Los sistemas de bases de datos estaban basados en el modelo jerárquico o en el de red. Estos dos primeros modelos están más íntimamente ligados a la implantación física de la base de datos que el modelo relacional.

El modelo de datos relacional representa a la base de datos como un conjunto de tablas. Aunque las tablas son un concepto simple e intuitivo, existe una correspondencia directa entre el concepto de una tabla y el concepto matemático de una relación.

II.4.1 ESTRUCTURA DE LAS BASES DE DATOS RELACIONALES.

Una base de datos relacional consiste en un conjunto de tablas que tienen asignado un nombre único. Un renglón de una tabla representa una relación entre un conjunto de valores. Puesto que una tabla es un conjunto de estas relaciones, existe una correspondencia entre el concepto de tabla y el concepto matemático de relación, del cual proviene el nombre de modelo relacional.

La siguiente figura nos muestra un ejemplo de una tabla llamada depósito :

En general, una tabla de n columnas debe ser un subconjunto de:

$$\begin{matrix} n \\ X \times D_1 \times \dots \times D_n \end{matrix}$$

Matemáticamente una relación se define como un subconjunto de un producto cartesiano de un conjunto de dominios. Debido a que las tablas son básicamente relaciones, se utilizan los términos matemáticos relación y tupla en vez de los términos tabla y columna.

En la relación de la figura anterior hay siete tuplos. Sea la variable de tuplo t el primer tuplo de la relación, se utiliza la notación $t(\text{nombre-sucursal})$ para denotar el valor en t del atributo nombre-sucursal. Puesto que una relación es un conjunto de tuplos, se utiliza la notación matemática $t \in r$ para indicar que el tuplo t está en la relación r .

Cuando se habla de una base de datos debe diferenciarse entre el esquema de base de datos, es decir, el diseño lógico de ésta y una instancia de la base de datos, que es la información contenida en ella. Así, una variable en los lenguajes de programación corresponde al concepto de una instancia de una relación.

Se adopta la convención de usar minúsculas para los nombres de relaciones y la primera letra mayúscula para los nombres de los esquemas de relación. Así, Esquema-depósito indica el esquema de la relación depósito.

Esquema-depósito = (nombre-sucursal, número-cuenta,
nombre-cuentahabiente, saldo)

En general, el esquema de una relación es una lista de atributos y sus correspondientes dominios. El hecho de que depósito sea una relación con el esquema Esquema-depósito se expresa de la siguiente manera :

depósito (Esquema-depósito)

Cuando se desea definir los dominios se utiliza la siguiente notación:

(nombre-sucursal : cadena, número-cuenta : entero,
 nombre-cuentahabiente : cadena, saldo : entero).

Supondremos que los esquemas de relaciones para la empresa son los siguientes :

Esquema-sucursal = (nombre-sucursal, activo, ciudad-sucursal)

Esquema-ctahab = (nombre-ctahab, calle, ciudad-ctahab)

Esquema-depósito=(nombre,sucursal,número-cta,nombre-ctahab,saldo)

Esquema-préstamo=(nombre-sucursal, número-préstamo, nom-ctahab, importe).

La figura siguiente muestra un ejemplo de la relación préstamo :

NOMBRE-SUC.	#-PRESTAMO	NOM.-CTAHAB.	IMPORTE
CENTRO	17	ESPINOZA	1000
SATELITE	23	RIVERA	2000
PERISUR	15	CASAS	1500
CENTRO	14	VIVEROS	500
COYOACAN	93	CANTU	900
VILLA NORTE	11	MONROY	1200
PORTALES	29	HERNANDEZ	1300
LUCERNA	16	AGUILAR	2000

RELACION " PRESTAMO "

Sea R un esquema de relaciones. Si se dice que un subconjunto K de R es una superllave de R , se van a considerar unicamente las relaciones $r(R)$ en las que no haya dos tuplas distintas que tengan los mismos valores para todos los atributos de K , es decir, si t_1 y t_2 estan en r y $t_1 \neq t_2$, entonces $t_1[K] \neq t_2[K]$.

LENGUAJES DE CONSULTA FORMALES.

Un lenguaje de consulta sirve para que el usuario solicite información de la base de datos. Estos lenguajes son normalmente de alto nivel, mayor que los lenguajes estándares de programación. Los lenguajes de consulta pueden clasificarse en lenguajes de procedimientos o sin procedimientos. En un lenguaje de procedimientos el usuario le ordena al sistema que realice una serie de operaciones con la base de datos para obtener el resultado deseado. En un lenguaje sin procedimientos, el usuario describe la información que desea sin indicar un procedimiento específico para obtenerla.

II.4.2 EL ALGEBRA RELACIONAL.

El Algebra relacional es un lenguaje de consulta de procedimientos. Existen cinco operaciones fundamentales en el algebra relacional, que son: selección, proyección, producto-cartesiano, unión y diferencia de conjuntos, todas ellas producen como resultado una nueva relación.

Además de las cinco operaciones principales, se van a introducir otras operaciones, a saber, intersección-conjuntos, producto (\otimes) tetha, producto natural y división. Estas se definirán en términos de las operaciones fundamentales.

Operaciones Fundamentales.

Las operaciones selección y proyección se denominan operaciones unarias, ya que actúan sobre una sola relación. Las otras tres relaciones operan sobre parejas de relaciones, por lo que se llaman operaciones binarias.

La selección opta por tuplas que satisfacen cierto predicado. Se utiliza la letra griega sigma minúscula (σ) para denotar la selección. El predicado aparece como subíndice de σ . La relación que constituye el argumento se da entre paréntesis después de la σ . Así, para elegir las tuplas de la relación préstamo en la que la sucursal es "Perisur", se escribe :

(prestamo)
 $\sigma_{\text{nombre-sucursal} = \text{"Perisur"}}$

En general, se permiten las comparaciones utilizando =, \neq , <, >, \leq , \geq , en el predicado de la selección. Además, pueden combinarse varios predicados para formar un predicado mayor utilizando los conectivos \wedge (y) y \vee (o). Así para hallar las tuplas que correspondan a préstamos mayores de xxx cantidad que ha hecho la sucursal "Perisur", se escribe :

(prestamo)
 $\sigma_{\text{nombre-sucursal} = \text{"Perisur"} \wedge \text{importe} > \text{xxx}}$

El predicado de la selección puede incluir comparaciones entre dos atributos. Para ilustrar esto se emplea el esquema de relaciones :

Esquema-cliente = (nombre-ctahab, nombre-empleado)

que indica que el empleado es el "banquero personal" del cuentahabiente. La relación cliente (Esquema-cliente) se observa en la siguiente figura.

NOMBRE-CTAHAB	NOM-EMPLEADO
NONROY GOMEZ NAVARRO	NAVARRO COLIN NAVARRO

RELACION CLIENTE

Pueden encontrarse todos los cuentahabientes que tienen el mismo nombre que su banquero personal, si se escribe

(cliente)

o nombre-ctahab = nombre-empleado

Si la relación cliente es como la de la figura anterior, la respuesta sería la relación mostrada a continuación.

NOMBRE-CTAHAB	NOM-EMPLEADO
NAVARRO	NAVARRO

Puede parecer redundante listar dos veces el nombre de la persona. Sería preferible una relación de un atributo (nombre-ctahab) que relacione a quienes tienen el mismo nombre de su banquero personal. La operación proyección permite producir esta relación. La operación proyección es unaria y copia su relación argumento quitando ciertas columnas. La proyección se señala con la letra griega pi (π) mayúscula. Como subíndice de π se pone una lista de todos los atributos que se desea que aparezcan en el resultado. La relación argumento se escribe después de π entre paréntesis.

Se regresará ahora a la pregunta: "Encontrar a los cuentahabientes que tienen el mismo nombre que su banquero personal". Se escribirá:

$$\pi_{\text{nombre-ctahab}}(\text{nombre-ctahabiente} = \text{nombre-empleado}(\text{cliente}))$$

Nótese que en vez de dar el nombre de una relación como argumento de la operación de proyección se dio una expresión que al evaluarse produce una relación.

Una operación que permite combinar información de varias relaciones es la de producto cartesiano, indicada por una cruz (\times). Esta operación es binaria, por lo que el producto cartesiano de las relaciones r_1 y r_2 se denotará como:

$$r_1 \times r_2$$

Supóngase que se requiere encontrar a todos los clientes del empleado bancario Navarro, así como su lugar de residencia. Se necesita información tanto de la relación cliente como de la relación cuentahabiente para lograrlo. Si la relación Cuentahabiente es como se muestra:

NOM-CTAHAB.	CALLE	CIUDAD-CTAHAB.
COLIN QUIROZ GOMEZ NAVARRO MONROY	LERMA NORTE LERMA ORIENTE AV. 6	SINALOA D.F. SINALOA PUEBLA D.F.

RELACION CUENTAHABIENTE

La relación resultante $r = \text{cliente} \times \text{cuentahabiente}$ sería como se muestra en la siguiente figura :

CLIENTE NOM-CTAHAB.	CLIENTE NOMBRE EMPLEADO	CUENTAHAB. NOM-CTAHAB.	CUENTAHAB. CALLE.	CUENTAHAB. CIUDAD-CTAHAB.
MONROY	NAVARRO	COLIN	LERMA	SINALOA
MONROY	NAVARRO	QUIROZ	NORTE	D.F.
MONROY	NAVARRO	GOMEZ	LERMA	SINALOA
MONROY	NAVARRO	NAVARRO	ORIENTE	PUEBLA
MONROY	NAVARRO	MONROY	AV. 6	D.F.
GOMEZ	COLIN	COLIN	LERMA	SINALOA
GOMEZ	COLIN	QUIROZ	NORTE	D.F.
GOMEZ	COLIN	GOMEZ	LERMA	SINALOA
GOMEZ	COLIN	NAVARRO	ORIENTE	PUEBLA
GOMEZ	COLIN	MONROY	AV. 6	D.F.
NAVARRO	NAVARRO	COLIN	LERMA	SINALOA
NAVARRO	NAVARRO	QUIROZ	NORTE	D.F.
NAVARRO	NAVARRO	GOMEZ	LERMA	SINALOA
NAVARRO	NAVARRO	NAVARRO	ORIENTE	PUEBLA
NAVARRO	NAVARRO	MONROY	AV. 6	D.F.

Es decir, simplemente se hace una lista de todos los atributos de las relaciones y se agrega el nombre de la relación de donde salió originalmente el atributo. Por lo tanto, todos los clientes de Navarro y la ciudades en las que viven se expresa :

(cliente X ctahab)
 σ cliente.nombre-empleado = "Navarro"

CLIENTE NON-CTAHAB.	CLIENTE NOMBRE EMPLEADO	CUENTAHAB NON-CTAHAB	CUENTAHAB. CALLE.	CUENTAHAB. CIUDAD-CTAHAB.
MORROY	NAVARRO	COLIN	LERMA	SINALOA
MORROY	NAVARRO	QUIROZ	MORTE	D.F.
MORROY	NAVARRO	GOMEZ	LERMA	SINALOA
MORROY	NAVARRO	CASAS	MORTE	D.F.
MORROY	NAVARRO	MERDOZA	PANQUE	PUEBLA
NAVARRO	NAVARRO	COLIN	LERMA	SINALOA
NAVARRO	NAVARRO	QUIROZ	MORTE	D.F.
NAVARRO	NAVARRO	CORZ	PANQUE	SINALOA
NAVARRO	NAVARRO	CASAS	LERMA	D.F.
NAVARRO	NAVARRO	MERDOZA	PANQUE	PUEBLA

RESULTADO DE CLIENTE X CUENTAHABIENTE.

Considérese ahora una consulta, "Encontrar a todos los cuentahabientes de la sucursal "Perisur". Es decir, se requiere buscar a todas las personas que tienen un préstamo, una cuenta o ambos. Todos los cuentahabientes que tienen un préstamo de la sucursal Perisur :

π nombre-ctahab (σ nombre-sucursal = "Perisur" (préstamo))

También se sabe cómo encontrar a todos los cuentahabientes que tienen una cuenta en la sucursal "Perisur" :

π nombre-ctahab (σ nombre-sucursal = "Perisur") (cuenta)

Por lo tanto, para contestar la consulta, se necesita la unión de estos dos conjuntos, es decir, todos los cuentahabientes que aparecen en cualquiera de las dos relaciones, o en ambas. Esto se logra mediante la operación binaria de unión, que se expresa como en la teoría de conjuntos con U. Por lo que la representación la unión se representa como :

$$\pi_{(prestamo)} ((nombre-ctahab (\sigma nombre-sucursal = "Perisur")) \cup \pi_{(deposito)} (nombre-ctahab (\sigma nombre-sucursal = "Perisur")))$$

La relación que resulta de esta consulta aparece en la siguiente figura :

NOMBRE-CTAHABIENTE
QUIROZ
CASAS
GOMEZ

El operador de diferencia de conjuntos, señalado por (-), permite encontrar las tuplas que están en una relación, pero no en otra. La expresión r - s resulta en una relación que contiene las tuplas que están en r pero no en s.

Pueden hallarse todos los cuentahabientes de la sucursal "Perisur" que tiene una cuenta allí y que no han sacado un préstamo en ella, escribiendo:

π
((nombre-ctahab (o nombre-sucursal = "Perisur") - (prestamo)))

π
(nombre-ctahab (o nombre-sucursal = "Perisur")) (deposito)

La relación resultante de esta consulta aparece en la siguiente figura:

NOMBRE-CTAHABIENTE
QUIROZ

II.5 NORMALIZACION.

Con el total de los datos a ser representados dentro de cualquier organización, surge el problema y la necesidad del diseño de la Base de Datos, esto es, decidir la estructura lógica para los datos, las relaciones necesarias y los atributos que deben contener.

La teoría de normalización forma parte importante del modelo relacional, ya que toma en cuenta el comportamiento de los datos.

Con algunas relaciones, actualizar datos puede tener consecuencias indeseables, las que se conocen como anomalías de la Base de Datos.

Un tipo de anomalías son las que surgen en la actualización de la información, es decir, problemas que se originan al añadir, borrar o cambiar tuplos en una relación. Existen también otras anomalías, llamadas de procesamiento, que son detectadas en la operación del modelo implantado y que afectan su rendimiento.

El proceso de normalización tiene como objetivo eliminar dichas anomalías al reorganizar las relaciones en forma normal.

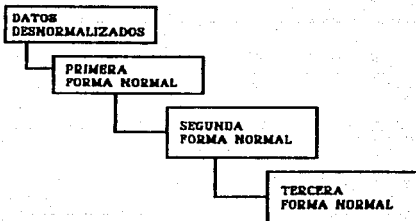
El término Normalización se refiere a la forma en que los datos son agrupados en registros. Tercera Forma Normal es un agrupamiento de datos diseñado para evitar las anomalías y problemas que puedan ocurrir con los datos.

Con la tercera forma normal, cada dato dentro de un registro hace referencia a una llave en particular que lo identifica en forma única. Tal llave puede estar compuesta por más de un dato o atributo. Cada dato del registro es identificado por la llave completa y no sólo por una parte de ésta. Ningún dato del registro es identificable por cualquier otro dato del mismo registro y que no sea parte de la llave.

La simplicidad de la tercera forma normal permite que los registros sean más fáciles de entender y cambiar, a diferencia de cuando los datos se organizan de una manera menos rigurosa. Formalmente agrupa los datos que están asociados a cada conjunto de entidades y separa los datos que pertenecen a otros conjuntos de entidades.

Los datos existen en la vida real en forma de grupos. Por ejemplo, en hojas de impuestos, licencias de manejo, etc. y por lo general, estos datos no están en una forma normalizada. La experiencia ha mostrado que cuando los datos se organizan en tercera forma normal las estructuras resultantes son más estables y se les puede hacer cambios más fácilmente. Es importante mencionar que la normalización describe la representación lógica y no física de los datos.

Para poner los datos en tercera forma normal se siguen tres pasos, que son los siguientes :



II.5.1 PRIMERA FORMA NORMAL (1FN)

La Primera Forma Normal se refiere a una colección de datos organizados en registros que no tienen grupos de datos repetidos. En otras palabras, estos registros son archivos o tablas de dos dimensiones.

Para explicar lo anterior, considérese el siguiente registro :

#EMPLEADO	NOM-EMPL	SEX	COD-EMPL	SALARIO	FECHA NAC.	HABILIDAD
					MM DD AA	COD-HAB. AÑOS-HAB.

El atributo "fecha-nacimiento" no causa problemas ya que ocurre solo una vez en cada registro, mientras que "habilidades" puede ocurrir varias veces en un mismo registro, por lo que éste no está en 1FN (no es un registro de dos dimensiones). Para normalizarlo, el grupo que se repite debe eliminarse y ponerse en un registro separado.

#EMPLEADO	NOM-EMPL	SEX	COD-EMPL	SALARIO	FECNA HAC
					MM DD AA

#EMPLEADO + CODIGO-HABILIDAD	AÑOS-HABILIDAD
------------------------------	----------------

El segundo registro tiene una llave concatenada "#empleado" + "código -habilidad". No se puede conocer "años-habilidad" (el número de años de experiencia que un empleado tiene en alguna actividad) a menos que se conozca "#empleado" (el número de empleado que hace referencia a éste) y "código-habilidad" (la habilidad en cuestión). En general, un archivo que no es de dos dimensiones se normaliza al convertirlo en dos o más archivos de dos dimensiones.

Cabe hacer énfasis que aquí se trata con la representación lógica y no física de los datos.

II.5.2 DEPENDENCIA FUNCIONAL.

Con el propósito de establecer las relaciones entre los datos, se debe analizar qué datos dependen de otros datos. La frase "funcionalmente dependiente" se define de la siguiente manera: "El atributo B de un registro R es funcionalmente dependiente del atributo A de R si, en cualquier instante de tiempo, cada valor de A no tiene más de un valor en B asociado con él en el registro R".

Decir que B es funcionalmente dependiente de A es equivalente a decir que A identifica a B. En otras palabras, si se conoce el valor de A se puede encontrar el valor de B asociado a A. Por ejemplo, en el siguiente registro, el atributo "salario" es funcionalmente dependiente de "#empleado". Para un "#empleado" existe un salario. Para encontrar el valor de salario en una base de datos normalmente se hace por medio de "#empleado". Este último es una llave que identifica al atributo "salario".

Se muestra "#emplado" como llave del siguiente registro :

#EMPLEADO	NOMBRE EMPLEADO	SALARIO	#PROYECTO	FECHA-TERMINACION
-----------	-----------------	---------	-----------	-------------------

Para representar las dependencias funcionales se usará una línea con una barra vertical :

#empleado ————— | salario

Esto indica que una instancia de salario está asociada con cada #empleado.

De esta manera, las dependencias funcionales para el registro anterior son:

"empleado" ————— | es dependiente de "nombre-emplado"
 "nombre-emplado" ————— | es dependiente de "#empleado"

"salario" — es dependiente de "nombre-empleado" o "#emplado"
 "#proyecto" — es dependiente de "nombre-empleado" o "#emplado"
 "fecha-terminación" — es dependiente de "nombre-empleado" o
 "#emplado" o "#proyecto"

"#emplado" no es funcionalmente dependiente de "salario" ya que más de un empleado puede tener el mismo salario. Se pueden dibujar estas dependencias funcionales de la siguiente manera :

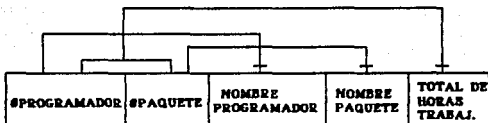


Un dato también puede ser funcionalmente dependiente de un grupo de datos y no tan solo depender de un único dato. Considerese, por ejemplo, el siguiente registro que muestra cómo los programadores invierten el tiempo.

#PROGRAMADOR	#PAQUETE	NOMBRE PROGRAMADOR	NOMBRE PAQUETE	TOTAL DE HORAS TRABAJ.
--------------	----------	--------------------	----------------	------------------------

"Total-hrs-Trabajadas" es funcionalmente dependiente de la llave concatenada "#programador" + "#paquete".

Las dependencias funcionales en este registro pueden ser ilustradas de la siguiente manera :



DEPENDENCIA FUNCIONAL COMPLETA.

Un dato o una colección de datos B de un registro R se dice que tiene dependencia funcional completa de otra colección de datos A del registro R, si B es funcionalmente dependiente de la totalidad de A y no sólo de un subconjunto de A.

Por ejemplo, en el registro anterior, "Total-hrs-Trabajadas" es completamente dependiente de la llave "#programador" + "#paquete" ya que ésta hace referencia a las horas que un programador ha trabajado en un determinado paquete. Ni "#programador solo, ni "#paquete solo identifican a "Total-hrs-Trabajadas"

"Total-hrs-Trabajadas" es el único dato que depende de la llave completa, mientras que "nombre-programador" depende sólo de "#programador" y "nombre-paquete" depende sólo de "#paquete".

II.5.3 SEGURIA FORMA NORMAL(2FN).

Una definición sencilla de 2FN es : "Cada atributo de un registro es funcionalmente dependiente de la llave completa del registro".

Cuando la llave está formada por más de un atributo, el registro puede no estar en 2FN. El registro anterior, cuya llave es "#programador" + "#paquete, no está en 2FN debido a que "Total-hrs-Trabajadas" depende de la llave completa, mientras que el "nombre-programador" y "nombre-paquete" dependen de sólo un atributo de la llave.

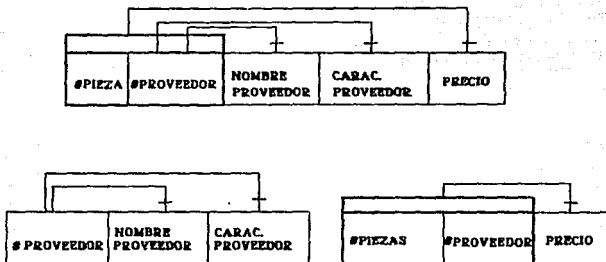
Similarmente, el siguiente registro no está en 2FN :

#PIEZAS	#PROVEEDOR	NOMBRE PROVEEDOR	CARAC PROVEEDOR	PRECIO
---------	------------	---------------------	--------------------	--------

Existen algunos problemas que pueden surgir con este registro :

- 1) No se pueden guardar detalles del proveedor hasta que éste suministre una parte.
- 2) Si un proveedor suministra muchas piezas se presenta redundancia de información.

Este tipo de irregularidades se eliminan si el registro se separa en dos, como se muestra a continuación :



Sólo precio depende de la llave completa, y los otros atributos se ponen en un registro aparte y que tiene como llave "número-proveedor".

En general, todos los atributos de un registro deben depender de la llave completa, en caso contrario deben ponerse en un registro aparte.

LLAVES CANDIDATO.

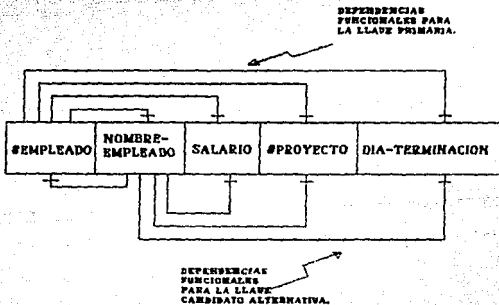
La llave de un registro normalizado debe cumplir con las siguientes prioridades :

- 1) Identificación Única. Para toda ocurrencia de un registro la llave debe identificarlo en forma única.

2) No redundancia. Ningún atributo que es parte de la llave puede ser eliminado sin eliminar la propiedad de identificación única.

En ocasiones ocurre que más de un atributo o conjunto de atributos pueden ser la llave de un registro. A éstas se les conoce como llaves candidato. Una de las llaves candidato debe ser elegida como llave primaria.

En el siguiente registro se representan las dependencias funcionales para la llave primaria y para la llave candidato alternativa.



En esta ilustración "nombre-empleado" es visto como una llave candidato (Una alternativa para "#empleada"). En la práctica esto no es hecho ya que los dos empleados pueden tener el mismo nombre. Sólo "#empleada" es verdaderamente única.

Una definición más completa de la Segunda Forma Normal es :

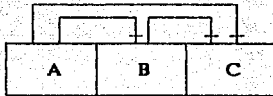
"Un registro R está en segunda forma normal si está en primera forma normal y todo atributo no llave depende funcionalmente de cada llave candidato de R".

II.5.4 TERCERA FORMA NORMAL (3FN).

Un registro que está en 2FN puede presentar otro tipo de anomalía: pueden existir atributos que no sean llaves pero que identifiquen a otros datos, lo que se conoce como "Dependencia Transitiva". La fase de la 3FN elimina este tipo de dependencias.

Supóngase que A, B, y C son tres atributos o conjuntos de atributos del registro R. Si C depende funcionalmente de B y B depende funcionalmente de A. Si el mapeo inverso no se da (si A no depende funcionalmente de B o B no depende funcionalmente de C), se dice que C depende transitivamente de A.

En el siguiente diagrama, C depende transitivamente de A si :



La conversión a 3FN elimina las dependencias transitivas al separar un registro en dos :



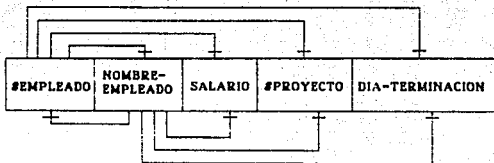
Una definición sencilla de la Tercera Forma Normal es :

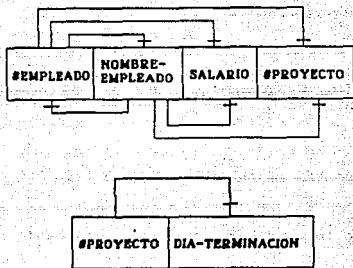
"Un registro está en 3FN y cada atributo no llave depende funcionalmente de la llave y sólo de la llave"

Una definición más formal y que incorpora a las llaves candidato es :

"Un registro R está en 3FN si está en 2FN y cada atributo no llave no depende transitivamente de cada llave candidato de R".

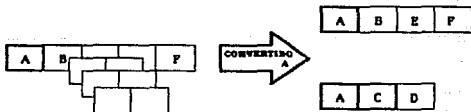
Lo anterior se ilustra en el siguiente ejemplo :





Los pasos a seguir para llegar a la 3FN se ilustran en el siguiente esquema :

CONVERSION A PRIMERA FORMA NORMAL (1F)



CONVERSION A SEGUNDA FORMA NORMAL



CONVERSION A TERCERA FORMA NORMAL



II.5.5 CONSIDERACIONES DE ALMACENAMIENTO Y EFICIENCIA.

El concepto de la Tercera Forma Normal es aplicable a todas las bases de datos.

Algunas objeciones son en el sentido de que requiere más almacenamiento o más tiempo de máquina. Usualmente, una estructura en 3FN tienen más archivos después del proceso descrito anteriormente. Sin embargo, y a pesar de usar más archivos, casi siempre se usa menos almacenamiento. La razón es que los registros que no están en 3FN generalmente tienen mucha redundancia.

En cuanto al tiempo de máquina y tiempo de acceso, frecuentemente son menores después de la normalización. Antes de la normalización muchos aspectos de los datos son puestos juntos y deben ser leídos todos a la vez. Después de la normalización se encuentran separados, de tal forma que sólo se lee un registro pequeño.

Asimismo, ya que en 3FN se tiene menos redundancia, el actualizar es más fácil.

Existen, sin embargo, excepciones a lo anterior. En ocasiones se debe diseñar registros que no estén en 3FN por razones de eficiencia.

CAPITULO III

D I S E Ñ O

E

I M P L E M E N T A C I O N

III.1 METODO DE LA RUTA CRITICA APLICADO AL PROYECTO.

Las actividades del sistema son :

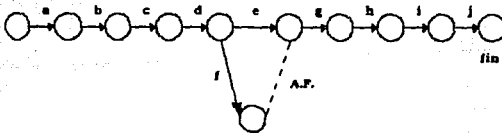
- a. - Especificación de requerimientos.
- b. - Recopilación de información.
- c. - Análisis de información.
- d. - Programación.
- e. - Pruebas.
- f. - Depuración.
- g. - Elaboración de Manuales.
- h. - Instalación del sistema.
- i. - Liberación.
- j. - Capacitación.

MATRIZ DE SECUENCIAS.

ACTIVIDAD	SECUENCIA
Inicio	a
a	b
b	c
c	d
d	e, f
e	g
f	g
g	h
h	i
i	j
j	final

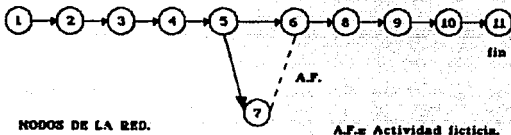
**MATRIZ DE
SECUENCIAS.**

Con esta matriz se construye la red de actividades, ejemplificada en la siguiente figura :

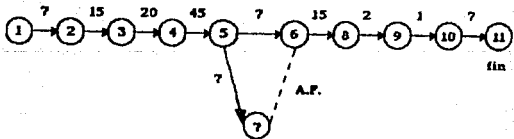


A.F. = Actividad ficticia.

numerar los nodos de la red. Esto permite identificar a las actividades como actividad (i, j).



Duración de las actividades :



La duración de las actividades está dada en días.

TIEMPO DE COMIENZO MAS CERCANO.

Cálculo del tiempo de comienzo más próximo o la fase llamada "Paso hacia adelante". El tiempo calculado se indica en un cuadro (a).

$$CP_1 = \text{Max} | 0 | = 0$$

$$CP_2 = \text{Max} | CP_1 + D_{s2} | = \text{Max} | 0 + 7 | = \text{Max} | 7 | = 7$$

$$CP_3 = \text{Max} | CP_2 + D_{s3} | = \text{Max} | 7 + 15 | = \text{Max} | 22 | = 22$$

$$CP_4 = \text{Max} | CP_3 + D_{s4} | = \text{Max} | 22 + 20 | = \text{Max} | 42 | = 42$$

$$CP_5 = \text{Max} | CP_4 + D_{s5} | = \text{Max} | 42 + 45 | = \text{Max} | 87 | = 87$$

$$CP_6 = \text{Max} | CP_5 + D_{s6} ; CP_5 + D_{s7} | = \text{Max} | 87 + 7 ; 87 + 7 | = \\ \text{Max} | 94 ; 94 | = 94$$

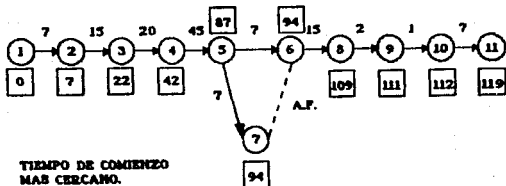
$$CP_7 = \text{Max} | CP_6 + D_{s8} | = \text{Max} | 94 + 15 | = \text{Max} | 109 | = 109$$

$$CP_8 = \text{Max} | CP_7 + D_{s9} | = \text{Max} | 109 + 2 | = \text{Max} | 111 | = 111$$

$$CP_9 = \text{Max} | CP_8 + D_{s10} | = \text{Max} | 111 + 1 | = \text{Max} | 112 | = 112$$

$$CP_{10} = \text{Max} | CP_9 + D_{s11} | = \text{Max} | 112 + 7 | = \text{Max} | 119 | = 119$$

$$CP_{10} = 119$$



TIEMPO DE TERMINACION MAS LEJANO

Calculo del tiempo de terminación más lejano ó la fase llamada "Paso hacia atrás". El tiempo calculado se indica en un rombo (◊).

$$TL_{10} = CP_{10} = 119$$

$$TL_9 = \text{Min} | TL_{10} - D_{9-10} | = \text{Min} | 119 - 7 | = 112$$

$$TL_8 = \text{Min} | TL_9 - D_{8-9} | = \text{Min} | 112 - 1 | = 111$$

$$TL_7 = \text{Min} | TL_8 - D_{7-8} | = \text{Min} | 111 - 2 | = 109$$

$$TL_6 = \text{Min} | TL_7 - D_{6-7} | = \text{Min} | 109 - 15 | = 94$$

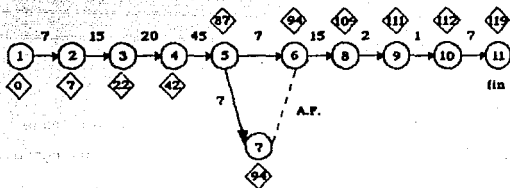
$$TL_5 = \text{Min} | TL_6 - D_{5-6} ; TL_6 - D_{5-7} | = \text{Min} | 94 - 7 ; 94 - 7 | = 87$$

$$TL_4 = \text{Min} | TL_5 - D_{45} | = \text{Min} | 87 - 45 | = 42$$

$$TL_3 = \text{Min} | TL_4 - D_{34} | = \text{Min} | 42 - 20 | = 22$$

$$TL_2 = \text{Min} | TL_3 - D_{23} | = \text{Min} | 22 - 15 | = 7$$

$$TL_1 = \text{Min} | TL_2 - D_{12} | = \text{Min} | 7 - 7 | = 0$$



TIEMPO DE TERMINACION
MAS LEJANO.

CALCULO DE LA HOLGURA TOTAL.

$$HT_{ij} = T_j - CP_i - D_{ij}$$

$$HT_{(1,2)} = TL_2 - CP_1 - D_{12} = 7 - 0 - 7 = 0$$

$$HT_{(2,3)} = TL_3 - CP_2 - D_{23} = 22 - 7 - 15 = 0$$

$$HT_{(3,4)} = TL_4 - CP_3 - D_{34} = 42 - 22 - 20 = 0$$

$$HT(4,5) = TL_5 - CP_4 - D_{45} = 87 - 42 - 45 = 0$$

$$HT(5,6) = TL_6 - CP_5 - D_{56} = 94 - 87 - 7 = 0$$

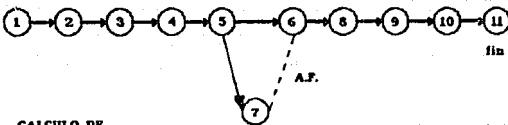
$$HT(6,7) = TL_7 - CP_6 - D_{67} = 94 - 87 - 7 = 0$$

$$HT(6,8) = TL_8 - CP_6 - D_{68} = 109 - 94 - 15 = 0$$

$$HT(8,9) = TL_9 - CP_8 - D_{89} = 111 - 109 - 2 = 0$$

$$HT(9,10) = TL_{10} - CP_9 - D_{9,10} = 112 - 111 - 1 = 0$$

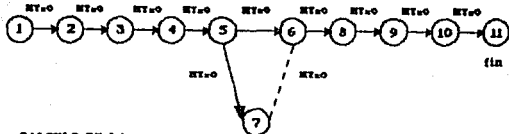
$$HT(10,11) = TL_{11} - CP_{10} - D_{10-11} = 119 - 112 - 7 = 0$$



**CALCULO DE
LA RUTA CRITICA.**

CALCULO DE LA RUTA CRITICA.

Todas las actividades con holgura total igual a cero se encuentran dentro de la Ruta Critica.



CALCULO DE LA HOLGURA TOTAL

La siguiente tabla resume todos los cálculos necesarios para determinar la Ruta Critica.

ACTIVIDAD (I, J)	DURACION (D _{IJ})	MAS PROXIMO		MAS LEJANO		HOLGURA	
		COMIENZO CPI □	TERM. TFI J	COMIENZO CII J	TERM. TFI J	TOTAL HTI J	LIBRE HTL J
1, 2	5	0	7	0	7	0 ^o	
2, 3	15	7	22	7	22	0 ^o	
3, 4	20	22	42	22	42	0 ^o	
4, 5	45	42	87	42	87	0 ^o	
5, 4	7	87	94	87	94	0 ^o	
5, 7	7	87	94	87	94	0 ^o	
6, 8	19	94	109	94	109	0 ^o	
8, 9	2	109	111	109	111	0 ^o	
9, 10	1	111	112	111	112	0 ^o	
10, 11	7	112	119	112	119	0 ^o	

• ACTIVIDAD CRITICA.

III.2 DISEÑO DE LA BASE DE DATOS.

IDENTIFICACION DE ENTIDADES Y ATRIBUTOS.

Como se mencionó con anterioridad, una entidad es un objeto que existe y que puede distinguirse de otros. En el caso de nuestra empresa, para el problema del manejo de la información de cajeros automáticos, podemos identificar cuatro conjuntos de entidades, a saber:

- Conjunto de entidades cajeros.
- Conjunto de entidades sucursales.
- Conjunto de entidades regionales.
- Conjunto de entidades resolvidores.

Cada entidad es representada por un conjunto de atributos. Así, los posibles atributos, y sus correspondientes dominios, para cada conjunto de entidades son los siguientes.

Atributos y dominios para el conjunto de entidades cajeros.

Atributo	Dominio
num-atm	4000 ≤ num-atm ≤ 8999
tipo-atm	{ IBM, Diebold }
num-c-regional	00 ≤ num-c-regional ≤ 99
num-sucursal	000 ≤ num-sucursal ≤ 999
c-cómputo	{ CSJ, CB, ROMA }
cpu	{ S90C, S90E }
ncp	{ J50, J60, J80, Jd0, ro1, ro2, 570, S90, d70, d90 }
enlace	{ radio, lp, lc, satélite, microondas, fibra óptica }
modelo-ncp	{ 3725, 3745 }
lineset	6 caracteres alfanuméricos
pu	8 caracteres alfanuméricos
lu	6 caracteres alfanuméricos
respaldo-pu	8 caracteres alfanuméricos
respaldo-enlace	{ radio, lp, lc, satélite, microondas, fibra óptica }

Atributos y dominios para el conjunto de entidades sucursales.

Atributo	Dominio
num-sucursal	000 ≤ num-sucursal ≤ 999
nombre-sucursal	cadena de caracteres
tot-atms	0 ≤ tot/atms ≤ 9
num-c-regional	00 ≤ num-c-regional ≤ 99
coordinador	cadena de caracteres
horario-servicio	cadena de caracteres
teléfono	cadena de caracteres
fax	cadena de caracteres
calle	cadena de caracteres
número	cadena de caracteres
colonia	cadena de caracteres
delegación	cadena de caracteres
cp	numero de 5 digitos

Atributos y dominios para el conjunto de entidades regionales.

Atributo	Dominio
num-c-regional	$00 \leq \text{num-c-regional} \leq 99$
nombre-c-regional	cadena de caracteres
num-remota	$00 \leq \text{num-remota} \leq 99$
teléfono	cadena de caracteres
fax	cadena de caracteres

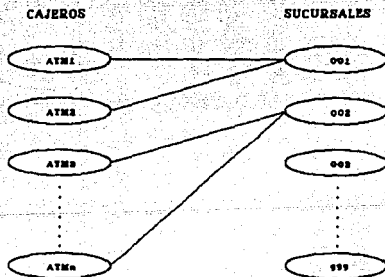
Atributos y dominios para el conjunto de entidades resolvedores.

Atributo	Dominio
nombre	cadena de caracteres
área	cadena de caracteres
aplicación	cadena de caracteres
extensión	cadena de caracteres
tel-particular	cadena de caracteres

MAPEO DE CARDINALIDAD.

El número de entidades con las que otra entidad puede asociarse se determina por medio del mapeo de cardinalidad, por lo que se procede a realizar el siguiente análisis de relaciones binarias.

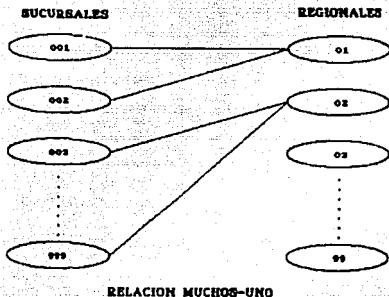
Mapeo de cardinalidad entre los conjuntos de entidades cajeros y sucursales.



RELACION MUCHOS-UNO

Ya que un cajero pertenece a una, y sólo una sucursal, y una sucursal puede tener uno o más cajeros automáticos, concluimos que este tipo de relación es muchos-uno.

Mapeo de cardinalidad entre los conjuntos de entidades sucursales y regionales.



Al igual que en el caso anterior, podemos ver que una sucursal pertenece a una, y sólo una remota, y que una remota puede tener uno a más sucursales, por lo que el tipo de relación también es muchos-uno.

Para el caso del conjunto de entidades resolvedores no es necesario realizar el análisis anterior, ya que su propósito es el proporcionar los nombres de las personas a quienes recurrir en caso de problemas con cualquier servicio de los que presta la institución y no guardan relación alguna con la información de cajeros automáticos. De hecho, éste es sólo un directorio de asesores.

DIAGRAMA E-R.

En base al análisis anterior y haciendo uso de los conceptos y convenciones para la elaboración de diagramas E-R, podemos expresar la totalidad de la estructura lógica de nuestra base de datos.

Como se puede observar en el diagrama E-R, la relación de cajeros y centros regionales es muchos-uno al igual que la relación cajeros y sucursales.

REDUCCION DEL DIAGRAMA E-R A TABLAS.

Nuestra base de datos conceptualizada por medio del diagrama E-R puede representarse por un conjunto de tablas (relaciones). Para cada conjunto de entidades existe una tabla única a la que se le asigna el mismo nombre del conjunto al que representa. Cada tabla contiene un número de columnas igual al número de atributos de las entidades y que también tienen nombres únicos.

Las tablas correspondientes a los conjuntos de entidades indicados anteriormente son las siguientes.

Tabla correspondiente al conjunto de entidades cajeros.

LU	TIPO-ATM	NUM-C-REG.	NUM-SUCURSAL	C-COMPUTO	CPU
----	----------	------------	--------------	-----------	-----

NCP	MODELO-NCP	ENLACE	RESP-ENLACE	LINESET	PU	RESP-PU
-----	------------	--------	-------------	---------	----	---------

CONJUNTO DE ENTIDADES CAJEROS

Tabla correspondiente al conjunto de entidades sucursales.

NUM-SUCURSAL	NUM-C-REGIONAL	NOM-SUCURSAL	TOT-ATM
--------------	----------------	--------------	---------

COORDINADOR	HORARIO-SERV.	TELEFONO	FAX	DIRECCION
				CALLE

DIRECCION				
NUMERO	COLONIA	CP	DELEGACION	POBLACION

Tabla correspondiente al conjunto de entidades regionales

NUM-C-REGIONAL	NOM-C-REGIONAL	TELEFONO	FAX
----------------	----------------	----------	-----

CENTRO REG. 1FN.

Tabla correspondiente al conjunto de entidades resolvedores.

NUM-RESOLVEDOR	NOM-RESOLVEDOR	AREA	EXTENSION
----------------	----------------	------	-----------

TEL-PARTICULAR

III.3 NORMALIZACION DE LA INFORMACION.

La normalización de la información es un agrupamiento de datos diseñado para evitar anomalías y problemas que puedan ocurrir con los datos.

Al normalizar la información, cada dato en un registro hace referencia a una llave que lo identifica en forma única. Cada dato es identificado por la llave completa y no sólo por una parte de la misma.

II.5.1 PRIMERA FORMA NORMAL (1FN).

Como se mencionó anteriormente, un registro está en 1FN si en éste no existen grupos de datos repetidos, es decir, un atributo no llave no debe tener valores diferentes para una misma llave.

● Conjunto de Entidades Cajeros.

LU	TIPO-ATM	NUM-C-REG.	NUM-SUCURSAL	C-COMPUTO	CPU
----	----------	------------	--------------	-----------	-----

NCP	MODELO-NCP	ENLACE	RESP-ENLACE	LINESET	PU	RESP-PU
-----	------------	--------	-------------	---------	----	---------

CONJUNTO DE ENTIDADES CAJEROS

La LU es un número que identifica en forma única a un y sólo un cajero, por lo que nunca tendrá asignados dos o más números para identificarlo.

El Tipo-Atm se refiere a la marca del cajero y nunca se da el caso en que una LU tenga dos tipos de cajero.

Núm-C-Regional es un número asignado a cada uno de los estados de la República Mexicana y un cajero está asignado a un sólo centro regional, es decir, una LU nunca se asigna a dos o más Centros Regionales.

Análogamente, los atributos C-Cómputo, CPU, NCP, Mod-NCP, Enlace, Resp-Enlace, Lineset, PU, Resp-PU son únicos, por lo que no tienen grupos de datos repetidos.

Así, el registro en IFN queda representado de la siguiente manera:

LU	TIPO-ATM	NUM-C-REG.	NUM-SUCURSAL	C-COMPUTO	CPU
----	----------	------------	--------------	-----------	-----

NCP	MODELO-NCP	ENLACE	RESP-ENLACE	LINESET	PU	RESP-PU
-----	------------	--------	-------------	---------	----	---------

• Conjunto de Entidades Sucursales

NUM-SUCURSAL	NUM-C-REGIONAL	NOM-SUCURSAL	TOT-ATM
--------------	----------------	--------------	---------

COORDINADOR	HORARIO-SERV.	TELEFONO	FAX	DIRECCION CALLE
-------------	---------------	----------	-----	--------------------

DIRECCION				
NUMERO	COLONIA	CP	DELEGACION	POBLACION

CONJUNTO DE ENTIDADES SUCURSALES

Cada sucursal tiene asignados un número y un nombre que la identifican, es decir, nunca tendrá asignados dos números o dos nombres distintos.

Asimismo, cada sucursal pertenece a un sólo Centro Regional.

Los atributos restantes también son únicos, no dándose el caso de tener grupos de datos repetidos, por lo tanto el registro está en 1FN.

• Conjunto de Entidades Regionales.

NUM-C-REGIONAL	NOM-C-REGIONAL	TELEFONO	FAX
----------------	----------------	----------	-----

CENTRO REG. IFN.

Los atributos número y nombre de Centro Regional son únicos y se corresponden biunivocamente, es decir, un Núm-C-Regional siempre identifica a un Nom-C-Regional y viceversa.

Los atributos teléfono y fax también son únicos, asumiendo que un C-Regional no tiene dos o más números telefónicos o de fax.

De tal forma que el registro se encuentra en IFN.

• Conjunto de Entidades Resolvedores.

NUM-RESOLVEDOR	NOM-RESOLVEDOR	EXTENSION	APLICACION
----------------	----------------	-----------	------------

TEL-PARTICULAR

Para cada resolvedor existe únicamente un número y un nombre asociados.

El área a la que pertenece siempre será una sola.

La aplicación que maneja puede ser una o más, por lo que el registro no está en IFN y es necesario ponerlo en un registro aparte.

Conjunto de Entidades Resolvedores.

NUM-RESOLVEDOR	NOM-RESOLVEDOR	AREA	EXTENSION
TEL-PARTICULAR			

Conjunto de Entidades Aplicaciones.

NUMERO RESOLVEDOR	APLICACION
-------------------	------------

CONJUNTO DE ENTIDADES "APLICACIONES".

SEGUNDA FORMA NORMAL (2FN)

Un registro está en 2FN si cada atributo no llave depende de la llave completa del registro. Para establecer esta dependencia es necesario analizar la relación existente entre los datos para conocer qué datos dependen de otros datos, lo que se conoce como dependencia funcional.

Usamos la notación siguiente :

A ————— B

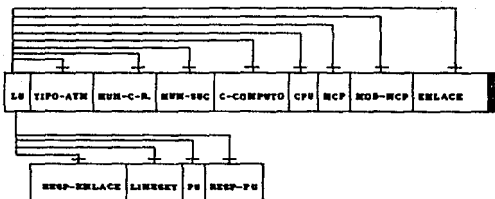
Para indicar que B depende funcionalmente de A.

• Conjunto de Entidades Cajeros.

Las dependencias funcionales para este registro son :

LU	—————	Tipo ATM
LU	—————	Núm-C-Regional
LU	—————	Núm-Bucursal
LU	—————	C-Computo
LU	—————	CPU
LU	—————	NCP
LU	—————	Mod-NCP
LU	—————	Enlace
LU	—————	Resp-Enalce
LU	—————	Lineset
LU	—————	PU
LU	—————	Resp-PU
PU	—————	Res-PU
Res-PU	—————	PU
NCP	—————	CPU
CPU	—————	C-Computo
PU	—————	Lineset
Lineset	—————	NCP
Resp-PU	—————	Lineset

Gráficamente se representa de la siguiente manera :



DEPENDENCIAS FUNCIONALES PARA ENTIDADES CAJEROS

Como llaves candidato se puede elegir a los atributos LU, PU, Resp-PU, NCP, CPU y Lineset. Sin embargo, se nota que todos los atributos del registro pueden ser identificados en forma única si se selecciona como llave primaria al atributo LU. El registro en 2FN es el siguiente :

LU	TIPO-ATM	NUM-C-REG.	NUM-SUCURSAL	C-COMPUTO	CPU
----	----------	------------	--------------	-----------	-----

NCP	MODELO-NCP	ENLACE	RESP-ENLACE	LINESET	PU	RESP-PU
-----	------------	--------	-------------	---------	----	---------

CONJUNTO DE ENTIDADES CAJEROS 2FN.

donde la llave primaria se indica en un recuadro con línea gruesa.

• Conjunto de Entidades Sucursales.

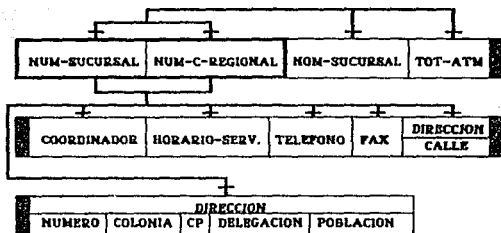
Dependencias funcionales :

Núm-Sucursal + Núm-C-Regional	—————	Nombre-Sucursal
Núm-Sucursal + Núm-C-Regional	—————	Tot-ATM's
Núm-Sucursal + Núm-C-Regional	—————	Coordinador
Num-Sucursal + Núm-C-Regional	—————	Horario Serv
Núm-Sucursal + Núm-C-Regional	—————	Teléfono
Núm-Sucursal + Núm-C-Regional	—————	Fax
Núm-Sucursal + Núm-C-Regional	—————	Dirección
Teléfono o Fax	—————	Núm-Sucursal
Teléfono o Fax	—————	Nombre-Sucursal
Teléfono o Fax	—————	Núm-C-Regional
Teléfono o Fax	—————	Tot-ATM's
Teléfono o Fax	—————	Coordinador
Teléfono o Fax	—————	Horario-Serv

Como llaves candidato se tiene a los atributos teléfono, fax y Núm-Sucursal + Núm-C-Regional.

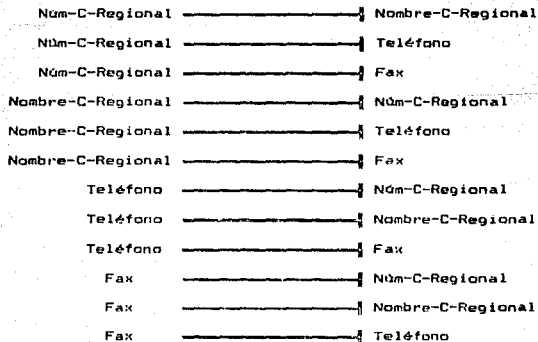
Teléfono y Fax pueden ser elegidos como llave primaria, empero no es práctico hacerlo ya que generalmente son datos que se requiere consultar. Seleccionar como llave primaria Núm-Sucursal + Núm-C-Regional es práctico y no causa problema para poner el registro en 2FN, en consecuencia se elige como llave primaria.

Gráficamente el conjunto de entidades sucursales.

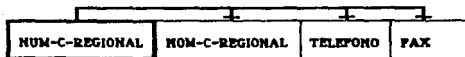


• Conjunto de Entidades Regionales.

Dependencias Funcionales :



En este caso, todos los atributos son llaves candidato. Por simplicidad y por ser práctico se elige como llave primaria Núm-C-Regional, cumpliéndose que los atributos no llave dependan de ésta.



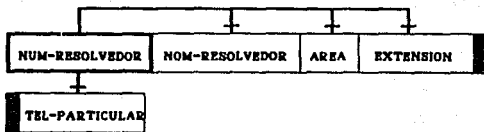
CENTRO REG. 2FN.

• Conjunto de Entidades Resolvedores.

Dependencias Funcionales.

Núm-Resolvedor	—————	Nombre-Resolvedor
Núm-Resolvedor	—————	Area.
Núm-Resolvedor	—————	Extensión
Núm-Resolvedor	—————	Tel-Particular
Tel-Particular	—————	Núm-Resolvedor
Tel-Particular	—————	Nombre-Resolvedor
Tel-Particular	—————	Area
Tel-Particular	—————	Extensión

Las llaves candidato son Núm-Resolvedor y Tel-Particular. Al igual que en el caso anterior, por simplicidad y por ser práctico, se elige como llave primaria Núm-Resolvedor, cumpliéndose que los atributos no llave dependan de ésta.



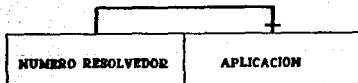
DEPENDENCIAS FUNCIONALES.

● Conjunto de Entidades Aplicaciones.

Dependencias Funcionales.

Núm-Resolvidor —————| Aplicación.

El registro en 2FN es :



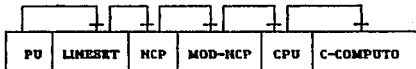
CONJUNTO DE ENTIDADES APLICACIONES GRAFICAMENTE EN 2FN.

TERCERA FORMA NORMAL (3FN)

En los registros anteriores puede ser que un atributo no llave o parte de una llave pueda identificar por si solo a otros atributos, esto se conoce como dependencia transitiva y se elimina al poner los registros en 3FN.

• Conjunto de Entidades Cajeros.

En este registro se tiene dependencia transitiva en los siguientes atributos no llave.



ENTIDADES CAJ. 3FN.

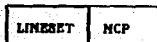
Por lo que es necesario eliminarla poniendo los atributos en registros separados.

Cajeros.

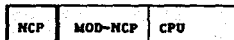
LU	TIPO-ATM	NUM-C-REG.	NUM-SUCURSAL	ENLACE
----	----------	------------	--------------	--------

RESP-ENLACE	LINESET	PU	RESP-PU
-------------	---------	----	---------

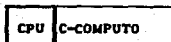
Lineset's



NCP's

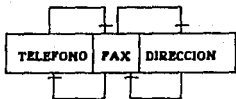


CPU's



● Conjunto de Entidades Sucursales.

Las dependencias para este registro son :



Fax depende funcionalmente de Teléfono y Dirección depende de Fax, pero como se da el caso inverso, es decir, Fax depende de Dirección y Teléfono de Fax, no existe dependencia transitiva, por lo que el registro está en 3FN.

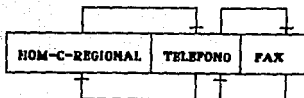
3FN.

NUM-SUCURSAL	NUM-C-REGIONAL	NOM-SUCURSAL	TOT-ATM
--------------	----------------	--------------	---------

COORDINADOR	HORARIO-SERV.	TELEFONO	FAX	DIRECCION
-------------	---------------	----------	-----	-----------

Conjunto de Entidades Regionales.

Las dependencias para este registro son :

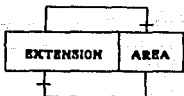


Teléfono depende del Nombre y Fax depende de Teléfono, y también se da el caso inverso, Teléfono depende de Fax y el Nombre depende de Teléfono, por lo que no existe dependencia transitiva y el registro está en 3FN.

NUM-C-REGIONAL	NOM-C-REGIONAL	TELEFONO	FAX
----------------	----------------	----------	-----

• Conjunto de Entidades Resolvedoras.

Las dependencias son :



No presentando dependencia transitiva. El registro en 3FN es:

Resolvedoras.

NUM-RESOLVEDOR	NOM-RESOLVEDOR	AREA	EXTENSION
TEL-PARTICULAR			

Conjunto de Entidades Aplicaciones.

En este registro no se presenta dependencia transitiva por lo que el registro está en 3FN.

NUMERO RESOLVEDOR	APLICACION
-------------------	------------

III.4 ESTRUCTURA DEL PROGRAMA

El sistema se compone de cinco módulos principales, de los que se derivan submódulos que, en conjunto, permiten acceder y manipular la información contenida en la base de datos.

MODULO 1. " CAJEROS "

Altas de Cajeros.

Bajas de Cajeros.

Modificaciones de Cajeros.

Consultas de Cajeros.

MODULO 2. " CONFIGURACIONES "

Altas de Configuraciones.

Bajas de Configuraciones.

Modificaciones de Configuraciones.

Consulta de Configuraciones.

Reportes de Configuraciones.

MODULO 3. " DIRECTORIO DE RESOLVEDORES "

Alta de Resolvedores.

Baja de Resolvedores.

Modificaciones de Resolvedores.

Consulta de Resolvedores.

MODULO 4. " REPORTES "

Información de Cajeros.

Directorio de Resolvedores.

MODULO 5. " UTILERIA "

Actualización de índices.

Se describe cada módulo especificando las diferentes formas en las que se puede consultar y obtener reportes de la información del componente a analizar.

MODULO 1. "CAJEROS"

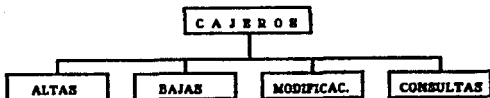
En este módulo estará contenida la información total de cajeros automáticos. Como se menciona anteriormente, está dividida en 4 incisos en los cuales se contempla lo siguiente :

Altas de ATM's . Se incluirá la información relacionada con la instalación de un nuevo ATM dentro de la red, con todos los parámetros que lo identifiquen y definan totalmente.

Bajas de ATM's . Borrar información de un ATM que ha sido excluido de la red o que sus parámetros de definición se han cambiado totalmente.

Módificaciones. Cuando existan cambios en los parámetros de definición tanto lógica como física de un ATM se requiere de actualizar la información, se accederá esta opción sin afectar ningún parámetro correspondiente a otro ATM.

Consulta de ATM's . Se podrá efectuar consulta de información acerca de un ATM basándose en el número de cajero.

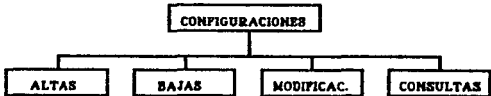


MODULO "1"
" CAJEROS "

MODULO 2. " CONFIGURACIONES "

Como se menciona en el capítulo I, una configuración es un formato en el cual se identifican las conexiones físicas hacia un ATM dentro de la red, mediante un conjunto de texto y gráficos. Se visualiza la conexión desde el Host hasta el punto remoto que es el ATM, especificando cada uno de los elementos e identificación.

El proceso para altas, bajas, modificaciones, consultas e impresión de configuraciones, se realizará por medio de un paquete gráfico llamado FONTASY.



MODULO "2"
" CONFIGURACIONES "

MÓDULO 3. " RESOLVEDORES "

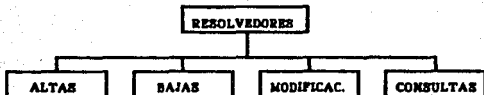
En este módulo se podrá manipular la información relacionada con las áreas y personas involucradas en la atención y solución a problemas de los Cajeros Automáticos, ya sea problemas de hardware como de software, y se contará con tres opciones, que se mencionan a continuación :

Altas de Resolvedores. Se incluirá la información relacionada con el resolvedor en cuestión, los datos más importantes y relevantes que nos puedan ayudar a localizarlos en caso de problemas graves dentro del sistema. por ejemplo :Nombre, teléfono, ubicación, extensión, área y aplicación.

Bajas de Resolvedores. Se borrará de la base de datos la información relacionada con el resolvedor en cuestión que por causas externas sea declarado fuera de la atención al servicio de cajeros automáticos.

Modificaciones. Actualizar o corregir información almacenada referida a un resolvedor, por ejemplo: Cambio de ubicación, de extensión telefónica, etc.

Consulta de Resolvedores. Acceso a información del directorio de resolvedores, ya sea por nombre o por una clave especial y única que lo identifica.



MODULO " 3 "

" RESOLVEDORES "

MODULO 4. " REPORTES "

En este módulo se podrá obtener impresiones de la información almacenada en la base de datos. Se tendrán los siguientes reportes :

INFORMACION DE ATM'S.

1.- Información de ATM's. Se obtendrán listados basados en :

Número de ATM. listado de los ATM's de la red.

Número y Nombre de la Sucursal. listado basado en el número y nombre de la sucursal con sus respectivos ATM's.

Identificación de NCP. listado de los ATM's configurados en un NCP de la red.

Centro regional. listado de los cajeros por centro regional.

Sucursales. listado con información de las sucursales.

4. - Reportes de Resolvedores.

Aquí obtendremos la impresión de formatos con la información correspondiente a las personas relacionadas con las aplicaciones vinculadas con el servicio de ATM's. Se tendrán reportes basados en el Área de trabajo, por la aplicación manejada, por su nombre y por clave.



MODULO "4" " REPORTE "

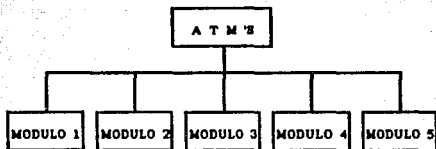
MODULO 5. " UTILERIA "

En esta opción se podrá realizar la actualización de índices para la base de datos.



MODULO " 5 " " UTILERIAS "

La siguiente gráfica nos muestra la estructura modular del sistema.



- 1.- CAJEROS
- 2.- CONFIGURACIONES
- 3.- DIRECTORIO
- 4.- REPORTES
- 5.- UTILERIA

MODULOS DE ACCESO AL SISTEMA DE CAJEROS AUTOMATICOS

ESTANDARES A SEGUIR DENTRO DEL SISTEMA.

En la programación se debe contar con ciertos estándares que permitan identificar y especificar la lógica de cada módulo que formará parte del sistema. Esto tiene la ventaja de hacer fácil la lectura y modificación de los módulos. El estándar a seguir en los módulos de programación es el siguiente :

Indicar el nombre del sistema.

Indicar la función del módulo.

Indicar módulo al que invoca.

Indicar módulo que lo invoca.

Los nombres de los módulos deben ser de, a lo máximo, 8 caracteres.

Los cuatro primeros caracteres indicarán el nombre de módulo.

Los otros cuatro caracteres harán referencia a la función que realiza el módulo.

También se deben incluir comentarios entre las líneas de programación que ayuden a la descripción de las acciones/funciones que realiza.

Por ejemplo: ATMSALTA.

ATMS.- Hace referencia al módulo.

ALTA.- Hace referencia a la función de registrar información correspondiente a un nuevo cajero automático.

III.5

CODIGO

FUENTE

CODIGO FUENTE.

III.5.1 MENU PRINCIPAL.

```
.....  
.....  
..... Proyecto : Control de Cajeros Automaticos. ....  
..... Rutina : Atms. ....  
..... Funcion : Invocar rutinas de inicializacion. ....  
..... Rutina que la invoca : Ninguna. ....  
..... Rutinas que invoca : Ambiente,logo,margen,mp. ....  
.....  
.....
```

```
.....  
***** Se invocan las rutinas para establecer SET'S del sistema. *****  
***** dibujar el logo, dibujar area de trabajo y el menu principal *****  
.....
```

```
DO ambiente  
DO logo  
DO margen  
DO mp  
RETURN
```

&& Regresa a DOS.


```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****      Rutina    : Margen.                               *****
*****      Funcion   : Dibujar recuadro de presentacion y   *****
*****                  area de trabajo.                     *****
*****      Rutina que la invoca : Atms.                     *****
*****      Rutina que invoca   : Ninguna.                   *****
*****

```

```

*****
*****      Variables.      *****
*****

```

```

borraestado = SPACE(78)
borratitulo = SPACE(78)
fecha       = DATE()
hora        = TIME()
titulo      = 'CONTROL DE CAJEROS AUTOMATICOS'
estado      = ' | Moverse | — Seleccionar | '
             '<Esc> Regresar | | F1 Ayuda | | '

```

```

*****
*****      Se dibuja recuadro de presentacion.      *****
*****      Imprime hora, nombre del sistema y la fecha.  *****
*****

```

```

CLEAR
SET COLOR TO bg/b,w/n
@ 00,00,22,79 BOX ' '
@ 01,01 SAY borratitulo
@ 02,01,02,78 BOX ' '
@ 03,01 SAY borratitulo
@ 23,01 SAY borraestado
@ 24,01 SAY borraestado
@ 23,00 SAY estado
SET COLOR TO g/r
@ 01,01 SAY hora
@ 01,09 SAY SPACE(16) + titulo + SPACE(16)
@ 01,71 SAY fecha
SET COLOR TO bg/b,w/n
RETURN

```

&& Regresa a rutina ATMS.

```

.....
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina      : MP.                                  *****
*****
*****      Funcion     : Menu principal.                      *****
*****
*****      Rutina que la invoca : Atm's.                      *****
*****
*****      Rutina que invoca   : Menuatms, menuconf,         *****
*****                          menurepo, menudir,           *****
*****                          indexa, ayudamp.              *****
*****
.....

```

```

.....
*****      Variables.      *****
*****
.....

```

```

borraestado = SPACE(80)
SET KEY 26 TO ayuda          && Asigna F1 como ayuda.

```

```

*****      Imprime menu principal.      *****
*****
.....

```

```

SET WRAP ON
MENU TO opcion
DO WHILE opcion <> 6
  @ 24,00 SAY borraestado
  @ 04,01 CLEAR TO 21,78
  SET MESSAGE TO 24 CENTER
  @ 03,03 PROMPT 'CAJEROS'          MESSAGE 'Altas Bajas'          +;
                                   'Modificaciones Consultas' +;
  @ 03,16 PROMPT 'CONFIGURACIONES' MESSAGE 'Altas Bajas'          +;
                                   'Modificaciones Consultas' +;
  @ 03,38 PROMPT 'REPORTES'       MESSAGE 'Reportes por ATM'      +;
                                   'Sucursal etc'                +;
  @ 03,52 PROMPT 'DIRECTORIO'     MESSAGE 'Altas Bajas'          +;
                                   'Modificaciones Consultas' +;
  @ 03,69 PROMPT 'UTILERIA'       MESSAGE 'Actualización de Indices'
MENU TO opcion

```

```

*****      Invoca rutina de la opcion seleccionada.      *****
*****
.....

```

```

DO CASE
CASE opcion = 1
  SET PATH TO c:\tesis\atms
  DO menuatms          && Invoca rutina para menu
                      && de cajeros.

```

```

CASE opcion = 2
  SET PATH TO c:\tesis\configur      && Invoca rutina para menu
  DO menuconf                       && configuraciones.
CASE opcion = 3
  SET PATH TO c:\tesis\reportes     && Invoca rutina para menu
  DO menurepo                       && de reportes.
CASE opcion = 4
  SET PATH TO c:\tesis\director     && Invoca rutina para menu
  DO menudir                         && de directorio.
CASE opcion = 5
  SET PATH TO c:\tesis\utileria     && Invoca rutina para la
  DO indexa                          && utileria.
OTHERWISE
  IF LASTKEY() = 27                 && Detecta si se ha presio-
    CLEAR                           && nado Esc.
    RETURN
  ENDIF
ENDCASE
ENDDO
RETURN                               && Regresa a DOS.

PROCEDURE ayuda
DO ayudamp                           && Rutina de ayuda para Fi.

```


III.5.2 MODULO CAJEROS.

```
*****  
*****  
***** Proyecto : Control de Cajeros Automaticos. *****  
***** Rutina : Menuatms. *****  
***** Funcion : Presentar opciones para manipular *****  
***** datos de cajeros. *****  
***** Rutina que la invoca : MP. *****  
***** Rutina que invoca : Atmsalta,Atmsbaja,Atmsmodi *****  
***** Atmscons,ayudatms. *****  
*****  
*****  
*****
```

***** ***** Variables. ***** *****

```
borraestado = SPACE(80)  
escape = 0  
alu = SPACE(04) && Variables del archivo Cajeros.  
anum_sucur = SPACE(03)  
anum_c_reg = SPACE(03)  
atipo_atm = SPACE(07)  
aenlace = SPACE(10)  
aresp_enla = SPACE(10)  
apu = SPACE(08)  
aresp_pu = SPACE(08)  
alineset = SPACE(08)  
afecha_lib = DATE()  
at_fs_acum = SPACE(06)  
ancp = SPACE(03)  
amod_ncp = SPACE(04)  
acpu = SPACE(04)  
ac_computo = SPACE(04)  
ames = SPACE(10)  
at_fs_mes = SPACE(06)  
anom_sucur = SPACE(15)  
atot_atms = SPACE(01)  
acoordinad = SPACE(30)  
ahora_serv = SPACE(15)  
atei_suc = SPACE(10)  
afax_suc = SPACE(10)  
pcalle = SPACE(15)  
anumero = SPACE(08)
```

&& Variables del archivo Sucursales.


```

@ 06,06 PROMPT 'BAJA DE CAJEROS' ' MESSAGE 'Borrar información'+
' de cajeros'
@ 07,06 PROMPT 'MODIFICACION CAJEROS' ' MESSAGE 'Corrección de datos'
@ 08,06 PROMPT 'CONSULTA DE CAJEROS' ' MESSAGE 'Consultar datos de'+
' cajeros'

```

```

MENU TO choice

```

```

***** Invoca rutina de la opción seleccionada. *****
*****

```

```

DO CASE
CASE choice = 1
DO atmsalta
escape = 1
SET COLOR TO bg/b,w/n
@ 04,01 CLEAR TO 21,78
&& invoca rutina para altas
&& de cajeros.
CASE choice = 2
DO atmsbaja
escape = 1
SET COLOR TO bg/b,w/n
@ 04,01 CLEAR TO 21,78
&& invoca rutina para bajas
&& de cajeros.
CASE choice = 3
DO atmsmodi
escape = 1
SET COLOR TO bg/b,w/n
@ 04,01 CLEAR TO 21,78
&& invoca rutina para modi-
&& ficaciones a cajeros.
CASE choice = 4
DO atmscons
escape = 1
SET COLOR TO bg/b,w/n
@ 04,01 CLEAR TO 21,78
&& invoca rutina para con-
&& sultas de cajeros.
ENDCASE
ENDDO
RETURN
&& Regresa a rutina MP.
PROCEDURE ayudal0
DO ayudatms
&& Rutina de ayuda para F1.

```



```

DO WHILE alta
bandlu      = .T.
bandsuc    = .T.
regsuc     = .T.
bandtipo   = .T.
bandenl    = .T.
bandrespe  = .T.
bandpu     = .T.
bandiset   = .T.
registset  = .T.
bandncp    = .T.
regncp     = .T.
bandmod    = .T.
bandcpu    = .T.
bandcomp   = .T.
@ 08,08 CLEAR TO 10,78
@ 24,00 SAY borraestado
@ 08,09 SAY 'LU :'
@ 08,13 GET alu
@ 08,18 SAY 'TIPO : '
@ 08,24 GET atipo_atm
@ 08,32 SAY 'ENLACE :'
@ 08,40 GET aonlance
@ 08,51 SAY 'RESPALDO :'
@ 08,61 GET aresp_enla
@ 09,09 SAY 'PU :'
@ 09,13 GET apu
@ 09,22 SAY 'RESPALDO :'
@ 09,32 GET aresp_pu
@ 09,41 SAY 'LSET :'
@ 09,47 GET alineset
@ 09,58 SAY 'FECHA LIB :'
@ 09,67 GET afecha_lib
READ
IF LASTKEY() = 27
RETURN
ENDIF

```

&& Se leen datos de cajeros.

PICTURE '9999'

PICTURE '!!!!!!!'

PICTURE '!!!!!!!!!!!!'

PICTURE '!!!!!!!!!!!!'

PICTURE '!!!!!!!!!!!!'

PICTURE '!!!!!!!!!!!!'

PICTURE '!!!!!!!!!!!!'

&& If para detectar si se ha presionado <Esc>.

```

*****
***** Se validan los datos leidos : lu, tipo, enlace, respaldo de *****
***** enlance, pu, respaldo de pu, lineset, etc. *****
*****

```

```

DO vallu WITH bandlu
IF bandlu
DO valtipo WITH bandtipo
IF bandtipo
DO valenlac WITH bandenl
IF bandenl
DO valrespe WITH bandrespe
IF bandrespe
DO valpu WITH bandpu
IF bandpu
aresp_pu = STRTRAN(apu,'N','R')
@ 09,32 SAY aresp_pu
DO valiset WITH bandiset, registset

```

```

DO CASE
CASE bandiset .AND. .NOT. regist
DO despncp
CASE .NOT. bandiset .AND. .NOT. regist
LOOP
CASE .NOT. bandiset .AND. regist
DO leenop
DO valncp WITH bandncp,regncp
DO CASE
CASE bandnop .AND. .NOT. regncp
DO despocu
CASE .NOT. bandnop .AND. .NOT. regncp
LOOP
CASE .NOT. bandnop .AND. regncp
DO leecpu
DO valmod WITH bandsod
IF bandmod
DO valcpu WITH bandocu
IF bandocu
DO valcomp WITH bandcomp
IF .NOT. bandcomp
LOOP
ENDIF
ELSE
LOOP
ENDIF
ELSE
LOOP
ENDIF
ENDIF
ENDCASE
ENDCASE

```

```

*****
**** Se chequea banderas para leer datos de la sucursal ****
*****

```

```

IF (bandiset .AND. .NOT. regist) .OR. ;
(.NOT. bandiset .AND. regist) .OR. ;
(bandncp .AND. .NOT. regncp) .OR. ;
(.NOT. bandncp .AND. regncp) ;
@ 10,54 SAY 'MSUC : ' ;
@ 10,60 GET anum_sucur PICTURE '999' ;
@ 10,64 SAY 'MC REG : ' ;
@ 10,72 GET anum_c_reg PICTURE '999' ;
READ

```

```

*****
**** Se valida los datos de la sucursal. ****
*****

```

```

DO valsuc WITH bandsuc, regsuc
DO CASE
CASE (bandsuc .AND. .NOT. regsuc)
DO altacaj && Se registran los datos.
213

```

```

@ 24,00 SAY borraestado
@ 24,pietitulo SAY mensajefin

fin = 0      && Loop para continuar en
             && altas.
DO WHILE fin <> 27 .AND. fin <> 13
  fin = INKEY()
  IF fin = 27
    alta = .F.
  ENDIF
ENDDO
DO inivar
CASE .NOT. bandsuc .AND. regsuc

DO altasucr  && Se leen datos de la suc.
DO altaobj  && Se registran los datos.

@ 24,00 SAY borraestado
@ 24,pietitulo SAY mensajefin

fin = 0      && Loop para continuar en
             && altas.
DO WHILE fin <> 27 .AND. fin <> 13
  fin = INKEY()
  IF fin = 27
    alta = .F.
  ENDIF
ENDDO
DO inivar
CASE .NOT. bandsuc .AND. .NOT. regsuc
LOOP
ENDCASE
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDDO
RESTSCREEN(04,01,21,78,pantatms)      && Se restablece pantalla.
SET COLOR TO n/bg,w/n                  && Regresa a rutina MENUATMS
RETURN
PROCEDURE ayudall                      && Rutina de ayuda.
DO ayudaita

```

```

.....
****
****      Proyecto : Control de Cajeros Automaticos.      ****
****
****      Rutina   : Inivar                                ****
****
****      Funcion  : Inicializar variables con valores nulos ****
****
****      Rutina que la invoca : Atmsalta, atmsbaja,      ****
****                          atmscons, atmsmodi.      ****
****
****      Rutinas que invoca   : Ninguna.                ****
****
.....

```

```

.....
****                          Variables                          ****
.....

```

```

alu          = SPACE(04)
anum_sucur  = SPACE(03)
anum_c_reg  = SPACE(03)
atipo_atm   = SPACE(07)
aoniace     = SPACE(10)
aresp_enla  = SPACE(10)
apu         = SPACE(08)
aresp_pu    = SPACE(08)
alineset    = SPACE(06)
afecha_lib  = DATE()
at_fs_acum  = SPACE(06)
ancp        = SPACE(03)
amod_ncp    = SPACE(04)
acpu        = SPACE(04)
ac_computo  = SPACE(04)
ames        = SPACE(10)
at_fs_mes   = SPACE(08)
anom_sucur  = SPACE(15)
atot_atms   = SPACE(01)
acoordinad = SPACE(30)
ahora_serv  = SPACE(15)
atel_suc    = SPACE(10)
afax_suc    = SPACE(10)
acalle      = SPACE(15)
anumero     = SPACE(08)
acolonias  = SPACE(20)
acp         = SPACE(05)
adelegacio  = SPACE(15)
aestado     = SPACE(15)
apoblacion  = SPACE(20)
anom_c_reg  = SPACE(20)
atel_reg    = SPACE(10)
afax_reg    = SPACE(10)

```

```

&& Variables del archivo
&& cajeros.

```

```

&& Variables del archivo
&& sucursales.

```

```

&& Variables del archivo
&& regional.

```

```
anum_resol  = SPACE(03)
anom_resol  = SPACE(30)
aextension  = SPACE(09)
aaplicacio  = SPACE(10)
atel_parti  = SPACE(07)
aarea       = SPACE(10)
RETURN
```

```
&& Variables del archivo
&& resolvedores.
```

```
&& Regresa a rutina atmsalta
&& atmsbaja,atmscon,atmsmodi
```

```

.....
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****      Rutina   : Vallu.                                *****
*****      Función  : Validar cuando un cajero ya          *****
*****                  ha sido de alta previamente.        *****
*****      Rutina que la invoca : Atasalta.                *****
*****      Rutina que invoca   : Mlue.                    *****
*****
*****
.....

```

```

.....
*****      Variables.      *****
.....

```

```

PARAMETER bandiu
bandiu = .T.

SELECT 1                                && Busca en archivo cajeros
SET ORDER TO 1                          && que el cajero no exista.
SEEK VAL(alu)
IF FOUND()
    DO mlue                               && Envia mensaje de error.
    bandiu =.F.
ENDIF
RETURN                                   && Regresa a rutina atasalta.

```



```

.....
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****      Rutina   : Valeniac.                               *****
*****      Funcion  : Validar enlace.                         *****
*****      Rutina que la invoca : Atmsalta, atmsmodi.        *****
*****      Rutinas que invoca   : Menine.                    *****
*****
.....

```

```

.....
*****      Variables.      *****
.....

```

```

PARAMETER banden1
banden1 = .T.

banden12 = .T.
barrastado = SPACE(80)
SELECT 8
SEEK aenlace
IF .NOT. FOUND()
    DO menine WITH banden12
    IF .NOT. banden12
        banden1 = .F.
ENDIF
ENDIF
RETURN

&& Busca en el archivo si existe el enlace.
&& Envia mensaje si enlace no existe.
&& Regresa a rutina atmsalta y atmsmodi.

```

```

*****
*****
***** Proyecto : Control de Cajeros Automaticos. *****
***** Rutina : Menine. *****
***** Funcion : Enviar mensaje cuando un enlace no *****
***** no ha sido de alta previamente. *****
***** Rutina que la invoca : Valenlac. *****
***** Rutina que invoca : Ninguna. *****
*****
*****
*****

```

```

*****
***** Variables. *****
*****
*****

```

```

PARAMETER banden12
banden12 = .T.

```

```

borraestado = SPACE(80)
mensaje = 'Enlace no registrado.    _ Registrar    <Esc> Corregir'

pistitulo = (79-LEN(mensaje))/2    && Calcula longitud del
@ 24,00 SAY borraestado            && mensaje.
@ 24,pistitulo SAY mensaje

salir = 0
DO WHILE salir <> 27 .AND. salir <> 13    && Loop para detectar Esc o
  salir = INKEY()                    && Enter.
  IF salir = 27
    banden12 = .F.
  ENDF
ENDDO
RETURN                                && Regresa a rutina valenlac

```



```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina   : Menrenln.                               *****
*****
*****      Funcion  : Enviar mensaje cuando un respaldo de *****
*****                  enlace no ha sido de alta previamente. *****
*****
*****      Rutina que la invoca : Valrespe.                   *****
*****
*****      Rutina que invoca   : Ninguna.                     *****
*****
*****
*****
*****

```

```

*****
*****      Variables.                                         *****
*****
*****

```

```

PARAMETER bandrespe2
bandrespe2 = .T.

```

```

borraestado = SPACE(80)
mensaje = 'Respaldo de enlace no registrado.    _ Registrar'
          '      <Eso> Corregir'

```

```

pietitulo = (79-LEN(mensaje))/2
@ 24,00 SAY borraestado
@ 24,pietitulo SAY mensaje
salir = 0
DO WHILE salir <> 27 .AND. salir <> 13
    salir = INKEY()
    IF salir = 27
        bandrespe2 = .F.
    ENDF
ENDDO
@ 24,00 SAY borraestado
RETURN

```

&& Calcula longitud del
 && mensaje.

 && Loop para detectar Esc o
 && Enter.

 && Regresa a rutina valrespe

```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina   : Valpu.                                *****
*****
*****      Funcion  : Validar cuando una PU ya              *****
*****                  ha sido de alta previamente.         *****
*****
*****      Rutina que la invoca : Atasalta.                 *****
*****
*****      Rutina que invoca   : Hpu.                       *****
*****
*****
*****
*****

```

```

*****
*****      Variables.                                     *****
*****
*****

```

```

PARAMETER bandpu
bandpu = .T.

```

```

SELECT 1                                && Busca en archivo si exis-
SET ORDER TO 2                          && te la pu.
SEEK apu
IF FOUND()                               && Envia mensaje de error.
    DO mpue
    bandpu = .F.
ENDIF
RETURN                                   && Regresa a rutina atsalta

```

```

*****
*****
****
****      Proyecto : Control de Cajeros Automaticos.      ****
****
****      Rutina   : Mpu.                                  ****
****
****      Funcion  : Enviar mensaje cuando una PU de     ****
****                  cajero ya ha sido asignada previamente. ****
****
****      Rutina que la invoca : Valpu.valpucon.          ****
****
****      Rutina que invoca   : Ninguna.                  ****
****
*****

```

```

*****
****                                     Variables.      ****
*****

```

```

borraestado = SPACE(80)
mensaje      = 'Unidad Fisica ya asignada.   < Esc > Continuar'

pietititulo = (79-LEN(mensaje))/2           && Calcula longitud del
@ 24,00 SAY borraestado                     && mensaje.
@ 24,pietitulo SAY mensaje
salir = 0
DD WHILE salir <> 27                          && Loop para detectar Esc.
  salir = INKEY()
ENDDO
RETURN                                         && Regresa a rutina valpu,
&& valpucon.

```



```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****      Rutina   : Valnop.                               *****
*****      Funcion  : Validar nop's.                       *****
*****      Rutina que la invoca : Atmsalta.                *****
*****      Rutinas que invoca  : Mopne.                    *****
*****
*****
*****

```

```

*****
*****      Variables.      *****
*****
*****

```

```

PARAMETER bandnop, regnop
bandnop = .T.
regnop = .T.

```

```

barraestado = SPACE(80)

```

```

@ 24,00 SAY barraestado

```

```

SELECT 3

```

```

SEEK anop

```

```

IF FOUND()

```

```

    bandnop = .T.

```

```

    regnop = .F.

```

```

ELSE

```

```

    bandnop = .F.

```

```

    DO mncpne WITH regnop

```

```

ENDIF

```

```

RETURN

```

```

&& Busca si el nop existe.

```

```

&& Envia mensaje si nop no

```

```

&& existe.

```

```

&& Regresa a rutina atmsalta

```



```

*****
*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****      Rutina   : Valmod.                               *****
*****      Funcion  : Validar cuando un modelo de nop no   *****
*****                  es correcta.                       *****
*****      Rutina que la invoca : Atmsalta,atmsmodi.       *****
*****      Rutina que invoca   : Mmodne.                  *****
*****
*****
*****

```

```

*****
*****      Variables.      *****
*****
*****

```

```

PARAMETER bandmod
bandmod = .T.

```

```

IF amod_nop <> '3745' .AND. amod_nop <> '3725'

```

```

    DO mmodne
    bandmod = .F.

```

```

    && Envia mensaje de error.

```

```

ENDIF
RETURN

```

```

    && Regresa a rutina atmsalta
    && atmsmodi.

```

```

*****
*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina   : Mmodne.                                *****
*****
*****      Funcion  : Enviar mensaje cuando un modelo      *****
*****                  de nop no es permitido.              *****
*****
*****      Rutina que la invoca : Valmod.                    *****
*****
*****      Rutina que invoca   : Ninguna.                    *****
*****
*****
*****
*****

```

```

*****
*****      Variables.      *****
*****
*****

```

```

borraestado = SPACE(80)
mensaje     = 'Modelo no existente.      < Esc > Continuar'

pietitulo = (79-LEN(mensaje))/2      && Calcula longitud del
@ 24,00 SAY borraestado              && mensaje.
@ 24,pietitulo SAY mensaje
salir = 0
DO WHILE salir <> 27                  && Loop para detectar Esc.
    salir = INKEY()
ENDDO
RETURN                                && Regresa a rutina valmod.

```

```

*****
*****
****
****      Proyecto : Control de Cajeros Automaticos.      ****
****
****      Rutina   : Valopu.                                ****
****
****      Funcion  : Validar cuando un opu no              ****
****                es correcto.                          ****
****
****      Rutina que la invoca : Atmsalta,atmsmodi.        ****
****
****      Rutina que invoca   : Mopune.                   ****
****
*****

```

```

*****
****      Variables.      ****
*****

```

```

PARAMETER bandcpu
bandcpu = .T.

IF acpu <> 'SBOC' .AND. mopu <> 'SBOF' .AND. mopu <> 'SBOE'
  .AND. acpu <> 'SBOD'
  DO mopune
  bandcpu = .F.
  && Envía mensaje de error.
ENDIF
RETURN
&& Regresa a rutina atmsalta
&& atmsmodi.

```



```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina   : Valcomp.                               *****
*****
*****      Funcion  : Validar cuando un centro de          *****
*****                  computo no es correcto.             *****
*****
*****      Rutina que la invoca : Atmsalta,atmsmodi.        *****
*****
*****      Rutina que invoca   : Mccompne.                  *****
*****
*****
*****
*****

```

```

*****
*****      Variables.      *****
*****
*****

```

```

PARAMETER bandcomp
bandcomp = .T.
IF ac_computo <> 'CSJ' .AND. ac_computo <> 'CB' ;
  .AND. ac_computo <> 'ROMA'
  DO mccompne                               && Envia mensaje de error.
  bandcomp = .F.
ENDIF
RETURN                                       && Regresa a rutina atmsalta
                                           && atmsmodi.

```

```

.....
.....
.....
Proyecto : Control de Cajeros Automaticos.
.....
Rutina : Mocompne.
.....
Funcion : Enviar mensaje cuando un centro de
computo no es permitido.
.....
Rutina que la invoca : Valccomp.
.....
Rutina que invoca : Ninguna.
.....
.....
.....

```

```

.....
.....
Variables.
.....
.....

```

```

borraestado = SPACE(80)
mensaje = 'Centro de cómputo no existente. < Esc > Continuar'

pittitulo = (79-LEN(mensaje))/2
@ 24,00 SAY borraestado
@ 24,pittitulo SAY mensaje
salir = 0
DO WHILE salir <> 27
    salir = INKEY()
ENDDO
RETURN

```

```

&& Calcula longitud del
&& mensaje.

&& Loop para detectar Esc.

&& Regresa a rutina valccomp

```



```

*****
*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina   : Despcpu.                               *****
*****
*****      Funcion  : Desplegar datos del cpu, modelo       *****
*****                  y centro de computo.                *****
*****
*****      Rutina que la invoca : Atmsalta.                 *****
*****
*****      Rutinas que invoca   : Ninguna.                  *****
*****
*****
*****

```

```

*****
*****      Variables.      *****
*****
*****

```

```

borraestado = SPACE(80)
@ 24,00 SAY borraestado
SELECT 3
SEEK anop
amod_nop = STR(mod_nop)
acpu = opu
SELECT 4
SEEK acpu
ac_computo = o_computo
@ 10,18 SAY 'MODELO : '
@ 10,28 GET amod_nop      PICTURE '!!!!'
@ 10,31 SAY 'CPU : '
@ 10,38 GET acpu         PICTURE '!!!!'
@ 10,41 SAY 'C COMP : '
@ 10,49 GET ac_computo  PICTURE '!!!!'
CLEAR GETS
RETURN
&& Busca el nop para obtener
&& el cpu.
&& Busca el cpu para obtener
&& el centro de computo.
&& Lee los datos.
&& Regresa a rutina atmsalta

```



```

@ 16,53 GET anuero          PICTURE '!!!!!!!'
@ 17,09 SAY 'COL : '
@ 17,14 GET acolonias      PICTURE '!!!!!!!!!!!!!!!!!!!!!!'
@ 17,35 SAY 'DEL : '
@ 17,40 GET adelegacio     PICTURE '!!!!!!!!!!!!!!!!!!!!!!'
@ 17,56 SAY 'CP : '
@ 17,60 GET acp            PICTURE '99999'
@ 18,09 SAY 'POBLACION : '
@ 18,20 GET apoblacion     PICTURE '!!!!!!!!!!!!!!!!!!!!!!'
READ
SELECT 7
SEEK anum_c_reg           && Se pide datos del centro
                           && regional.
IF .NOT. FOUND()
  @ 18,42 SAY 'C REG : '
  @ 18,49 GET anum_c_reg   PICTURE '!!!!!!!!!!!!!!!!!!!!!!'
  @ 19,09 SAY 'TEL : '
  @ 19,14 GET atel_reg     PICTURE '!!!!!!!!!!!!'
  @ 19,25 SAY 'FAX : '
  @ 19,30 GET afax_reg     PICTURE '!!!!!!!!!!!!'
  READ
ELSE
  anum_c_reg = nom_c_regi  && Se leen datos del centro
  atel_reg   = tel_reg     && regional.
  afax_reg   = fax_reg
  @ 18,42 SAY 'C REG : '
  @ 18,49 GET anum_c_reg   PICTURE '!!!!!!!!!!!!!!!!!!!!!!'
  @ 19,09 SAY 'TEL : '
  @ 19,14 GET atel_reg     PICTURE '!!!!!!!!!!!!'
  @ 19,25 SAY 'FAX : '
  @ 19,30 GET afax_reg     PICTURE '!!!!!!!!!!!!'
  CLEAR GETS
ENDIF
@ 24, pietitulo SAY mensaje && Loop para detectar Esc o
salir = 0 && Enter.
DO WHILE salir <> 27 .AND. salir <> 13
  salir = INKEY()
  IF salir = 13
    datook = .F.
  ENDIF
ENDDO
ENDDO
RESTSCREEN(04,01,21,78,pantsucs) && Se restablece pantalla de
SET COLOR TO n/bg,w/n && cajeros.
RETURN && Regresa a rutina atmsalta
&& atmsodi.
&& Invoca ayuda.
PROCEDURE ayudall1
DO ayudasuc

```



```
    @ 18,35 SAY '
    @ 19,35 SAY '
ENDIF
salir = 0
DO WHILE salir <> 27
  salir = INKEY()
ENDDO
RESTSCREEN(04,02,21,79,pante)
SET COLOR TO n/bg,w/n
RETURN
```

```
< Eso > Salir '
```

```
&& Loop para detectar Esc.
```

```
&& Se restablece pantalla.
```

```
&& Regresa a rutina altasucr
```



```

SELECT 3
APPEND BLANK
REPLACE nop WITH anop, mod_nop WITH VAL(amod_nop), cpu WITH acpu
SELECT 4
SEEK acpu
IF .not. FOUND()
  APPEND BLANK
  REPLACE cpu WITH acpu, c_computo WITH ac_computo
ENDIF
ENDIF
ENDIF

IF regsuc
SELECT 6
SEEK anum_sucur + anum_c_reg
IF .NOT. FOUND()
APPEND BLANK
REPLACE num_sucurs WITH anum_sucur, num_c_reg WITH anum_c_reg
REPLACE nom_sucurs WITH anon_sucur, total_atms WITH (total_atms + 1)
REPLACE coordinado WITH acoordinad, hora_serv WITH ahora_serv
REPLACE tel_suc WITH atel_suc , fax_suc WITH afax_suc
REPLACE calle WITH acalle , numero WITH anumero
REPLACE colonia WITH acolonia , cp WITH VAL(acp)
REPLACE delegacion WITH adelegacio, poblacion WITH apoblacion

SELECT 7
SEEK VAL(anum_c_reg)
IF .not. FOUND()
  APPEND BLANK
  REPLACE num_c_reg WITH anum_c_reg, nom_c_reg WITH anon_c_reg
  REPLACE tel_reg WITH atel_reg , fax_reg WITH afax_reg
ENDIF
ENDIF
ENDIF
RETURN

```

&& Se registran los datos
&& de la sucursal.

&& Se registran los datos
&& del centro regional.

&& Regresa a rutina atmsalta

```

*****
****
****      Proyecto : Control de Cajeros Automaticos.      ****
****
****      Rutina   : Ayudalta.                               ****
****
****      Funcion  : Proporcionar ayuda en la pantalla de   ****
****                  alta para cajeros.                   ****
****
****      Rutina que la invoca : Atsalta.                   ****
****
****      Rutina que invoca   : Ninguna.                    ****
****
*****

```

```
pante = SAVESCREEN(04,02,21,79)           && Salva pantalla de cajeros
```

```

SET COLOR TO bg/n
@ 05,35,21,78 BOX
SET COLOR TO n/bg,w/n
@ 04,34 TO 20,77
@ 05,35 SAY '                               Ayuda                               1/2 '
@ 06,35 SAY '
@ 07,35 SAY ' En esta pantalla se debe proporcionar la '
@ 08,35 SAY ' información requerida por los campos. Si '
@ 09,35 SAY ' se comete algún error use las flechas '
@ 10,35 SAY ' para posicionarse en el campo a corregir '
@ 11,35 SAY ' y ponga la información correcta. '
@ 12,35 SAY ' Las características de los campos son : '
@ 13,35 SAY ' LU : 4 dígitos numéricos . Ej. 4950 '
@ 14,35 SAY ' TIPO : sólo puede ser DIEBOLD ó IBM. '
@ 15,35 SAY ' ENLACE : radio, lp, lc, microondas, etc. '
@ 16,35 SAY ' RESPALDO : Es el respaldo del enlace y '
@ 17,35 SAY ' puede ser alguno de los valo- '
@ 18,35 SAY ' res para el enlace. '
@ 19,35 SAY ' <PgDw> Más... '
salir = 0

```

```

DO WHILE salir <> 27 .AND. salir <> 3           && Loop para detectar PgDwn.
    salir = INKEY()

```

```

ENDDO
IF salir = 3
@ 05,35 SAY '                               Ayuda                               2/2 '
@ 06,35 SAY '
@ 07,35 SAY ' PU : es la unidad física. Ej. PUI59N01 '
@ 08,35 SAY ' RESPALDO : es el respaldo de la PU. '
@ 09,35 SAY ' Ej. PUI59R01 '
@ 10,35 SAY ' LSET : es el lineset al que se encuentra '
@ 11,35 SAY ' conectada la PU. '
@ 12,35 SAY ' FECHA LIB : fecha de liberación del ATM. '
@ 13,35 SAY ' NCP : ctrl. de comunicaciones. Ej. j80 '
@ 14,35 SAY ' MODELO : modelo del NCP. Ej. 3745 '
@ 15,35 SAY ' CPU : procesador central. Ej. S90C '
@ 16,35 SAY ' C COMP : centro de cómputo. Ej. CSJ '

```

```

@ 17,35 SAY ' #SUC : sucursal que encargada del ATM. '
@ 18,35 SAY ' #C REG : estado de la república. '
@ 19,35 SAY ' < Esc > Salir '
ENDIF
salir = 0
DO WHILE salir <> 27
    salir = INKEY()
    && Loop para detectar Esc.
ENDDO
RESTSCREEN(04,02,21,79,pante)
SET COLOR TO n/bg,w/n
RETURN
&& Restablece pantalla de
&& cajeros.
&& Regresa a rutina atmsalta
```



```

apu          = pu
aresp_pu    = resp_pu
alineset    = lineset
afecha_lib  = fecha_libe
SELECT 6
SEEK anum_sucur + anum_o_reg          && Selecciona archivo de
anum_sucur = nom_sucours              && sucursales.
SELECT 7
SEEK anum_o_reg                      && Selecciona archivo de
anum_o_reg = nom_o_regi              && centros regionales.
@ 08,18 SAY 'TIPO i'
@ 08,24 GET atipo_atm PICTURE '!!!!!!!' && Presenta la informacion.
@ 08,32 SAY 'ENLACE i'
@ 08,40 GET aenlace PICTURE '!!!!!!!!!!'
@ 08,51 SAY 'RESPALDO i'
@ 08,61 GET aresp_enia PICTURE '!!!!!!!!!!'
@ 09,09 SAY 'PU i'
@ 09,13 GET apu PICTURE '!!!!!!!'
@ 09,22 SAY 'RESPALDO i'
@ 09,32 GET aresp_pu PICTURE '!!!!!!!'
@ 09,41 SAY 'LSET i'
@ 09,47 GET alineset PICTURE '!!!!!!!'
@ 09,58 SAY 'FECHA LIB i'
@ 08,87 GET afecha_lib
@ 10,09 SAY 'SUC i'
@ 10,14 GET anum_sucur PICTURE '!!!!!!!!!!!!!!!!!!!!'
@ 10,30 SAY 'C REG i'
@ 10,37 GET anum_o_reg PICTURE '!!!!!!!!!!!!!!!!!!!!!!!!!!!!'
CLEAR GETS
DO valbaja WHIT bandbaja              && Valida la baja.
IF bandbaja
DO bajacaj                            && Se eliminan los datos.
@ 24,00 SAY borraestado
@ 24,pietitulo SAY mensajesfin
fin = 0                                && Loop para detectar Esc.

DO WHILE fin <> 27 .AND. fin <> 13
  fin = INKEY()
  IF fin = 27
    baja = .F.
  ENDIF
ENDDO
ENDIF
ELSE
  IF LASTKEY() = 27                    && Detecta si se ha presio-
  RETURN                               && nado Esc.
  ENDIF
  DO mlune                             && Envia mensaje de error.
  ENDIF
ENDDO
RESTSCREEN(04,01,21,78,pantatms)     && Restablece pantalla mp.
SET COLOR TO n/bg,w/n                 && Regresa a rutina mp.
RETURN
PROCEDURE ayuda12                      && Invoca rutina de ayuda.
DO ayudabaj

```



```

*****
*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina   : Bajacaj.                               *****
*****
*****      Funcion  : Eliminar datos de los archivos.       *****
*****
*****      Rutina que la invoca : Atmsbaja.                  *****
*****
*****      Rutinas que invoca   : Ninguna.                   *****
*****
*****
*****

```

```

*****
*****      Variables.      *****
*****
*****

```

```

SELECT 1                                     && Selecciona archivo de
SET ORDER TO 1                               && cajeros.
SEEK alu
DELETE                                       && Borra la informacion.
PACK

SELECT 6                                     && Selecciona archivo de
SEEK anum_sucur + anum_c_reg                && sucursales.
REPLACE total_atms WITH (total_atms - 1)
IF total_atms <= 0
  DELETE                                     && Borra la informacion si
  PACK                                       && ya no tiene cajeros.
ENDIF
RETURN                                       && Regresa a rutina atmsbaja

```

```

*****
*****
***** Proyecto : Control de Cajeros Automaticos. *****
***** Rutina : Mlune. *****
***** Funcion : Enviar mensaje cuando se quiere dar *****
***** de baja un cajero que no existe. *****
***** Rutina que la invoca : Atmsbaja,atmsmodi,atmscons. *****
***** Rutina que invoca : Ninguna. *****
*****
*****
*****

```

```

*****
***** Variables. *****
*****
*****

```

```

borraestado = SPACE(80)
mensaje = 'Cajero no existe. < Esc > Continuar'
pietitulo = (79-LEN(mensaje))/2 && Calcula longitud del
&& mensaje.

@ 24,00 SAY borraestado
@ 24,pietitulo SAY mensaje
salir = 0
DO WHILE salir <> 27 && Loop para detectar Esc.
  salir = INKEY()
ENDDO
RETURN && Regresa a rutina atmsbaja
&& atmsmodi,atmscons.

```



```

*****
*****
***** Proyecto : Control de Cajeros Automaticos. *****
***** Rutina : Atmscons. *****
***** Funcion : Permitir la consulta de datos de *****
***** cajeros. *****
***** Rutina que la invoca : Menuatms. *****
***** Rutinas que invoca : Inivar, conslee, conscaj, *****
***** consuc, mluno, ayudacon. *****
*****
*****
*****
*****

```

```

*****
***** Variables. *****
*****

```

```

fin = 0
cons = .T.
mensaje = ' ' Continuar <Esc> Terminar'

```

```

pietitulo = (79-LEN(mensaje))/2
borraestado = SPACE(80)
pantatms = SAVESCREEN(04,01,21,78)
SET KEY 28 TO ayuda14
SET COLOR TO bg/n
@ 08,08,12,78 BOX ' ' |'-'|'''
SET COLOR TO n/bg,w/n
@ 07,07 TO 11,77
@ 08,08 CLEAR TO 10,78
@ 24,00 SAY borraestado
DO WHILE cons
DO inivar
@ 08,08 CLEAR TO 10,78
@ 24,00 SAY borraestado
@ 08,09 SAY 'LU : '
@ 08,13 GET alu PICTURE '9999'
READ
SELECT 1
SET ORDER TO 1
SEEK alu
IF FOUND()
DO conslee
DO conscaj
pantcons = SAVESCREEN(13,07,21,78)
SET COLOR TO bg/n
@ 14,08,21,78 BOX ' ' |'-'|'''

```

&& Calcula longitud del
&& mensaje.
&& Salva pantalla menuatms.
&& Asigna F1 como ayuda.

&& Loop para consultas.

&& Se lee dato del cajero a
&& consultar.

&& Busca el cajero.

&& Obtiene informacion de
&& los cajeros.
&& Presenta la informacion.
&& Salva pantalla de con-
&& sulta.

```

SET COLOR TO n/bg,w/n
@ 13,07 CLEAR TO 20,78
@ 13,07 TO 20,77
DO consuc                                && Presenta la informacion
CLEAR GETS                                && de sucursales.
@ 24,00 SAY borraestado !
@ 24, pietitulo SAY mensajefin
fin = 0
DO WHILE fin <> 27 .AND. fin <> 13      && Loop para detectar Esc o
  fin = INKEY()                          && Enter.
  IF fin = 27
    cons = .F.
 ENDIF
ENDIF
ENDDO
RESTSCREEN(13,07,21,78,pantoons)         && Restablece pantalla de
ELSE                                       && consulta.
  IF LASTKEY() = 27                       && Detecta se se ha pulsado
    RETURN                                 && Esc.
  ENDIF
  DO mlune                                 && Envía mensaje de error.
ENDIF
ENDDO
RESTSCREEN(04,01,21,78,pantatms)         && Restablece pantalla de
SET COLOR TO n/bg,w/n                   && menuatms.
RETURN                                    && Regresa a rutina atmsodi
PROCEDURE ayuda14                         && Invoca ayuda.
DO ayudacon

```



```
acp      = str(cp)
adelegacio = delegacion
aestado  = estado
apoblacion = poblacion
SELECT 7
SEEK anum_c_reg
anom_c_reg = nom_c_regi
atel_reg = tel_reg
afax_reg = fax_reg
RETURN
```

```
&& Se obtiene informacion de
&& centros regionales.
```

```
&& Regresa a rutina atmmodi
&& atmcons.
```



```

.....
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****      Rutina   : Ayudacion.                            *****
*****      Funcion  : Proporcionar ayuda en la pantalla de  *****
*****                  consulta de cajeros.                *****
*****      Rutina que la invoca : Atmscons.                  *****
*****      Rutina que invoca   : Ninguna.                   *****
*****
.....

```

```

.....
*****                               Variables.                               *****
.....

```

```

salir = 0
pante = SAVESCREEN(04,02,21,79)      && Salva pantalla atmscons
SET COLOR TO bg/n
@ 05,35,21,78 BOX                    ' |J-4|'
SET COLOR TO n/bg,w/n
@ 04,34 TO 20,77
@ 05,35 SAY '                          Ayuda                          1/1 '
@ 08,35 SAY '
@ 07,35 SAY ' Proporcione el número de cajero del que
@ 08,35 SAY ' se desea consultar información.
@ 09,35 SAY ' El valor debe ser numérico. Ej. 4950
@ 10,35 SAY '
@ 11,35 SAY '
@ 12,35 SAY '
@ 13,35 SAY '
@ 14,35 SAY '
@ 15,35 SAY '
@ 16,35 SAY '
@ 17,35 SAY '
@ 18,35 SAY '
@ 19,35 SAY '
salir = 0
DO WHILE salir <> 27                  && Loop para detectar Esc.
    salir = INKEY()
ENDDO
RESTSCREEN(04,02,21,79,pante)      && Restablece pantalla.
SET COLOR TO n/bg,w/n
RETURN                               && Regresa a rutina atmsmodi

```



```

DO Inivar
bandtipo = .T.
bandenl = .T.
bandrespe = .T.
bandpu = .T.
bandlset = .T.
reglset = .T.
bandncp = .T.
regncp = .T.
bandmod = .T.
bandcpu = .T.
bandcomp = .T.
bandsuc = .T.
regsuc = .T.
@ 08,08 CLEAR TO 10,78
@ 24,00 SAY borraestado
@ 08,09 SAY 'LU :'
@ 08,13 GET alu PICTURE '9999'
READ
SELECT 1
SET ORDER TO 1
SEEK alu
IF FOUND()
DO conslee
DO conscaje
READ

```

&& Se inicializa variables.
 && Lee cajero a modificar.
 && Selecciona archivo de cajeros.
 && Obtiene la informacion de los archivos.
 && Presenta los datos a modificar.

```

*****
***** Se validan los datos : tipo, enlace, respaldo de enlace *****
***** pu, lineset, ncp, cpu, etc. *****
*****

```

```

DO valtipo WITH bandtipo
IF bandtipo
DO valenlac WITH bandenl
IF bandenl
DO valrespe WITH bandrespe
IF bandrespe
DO valpucon WITH bandpu
IF bandpu
aresp_pu = STRTRAN(apu,'N','R')
@ 09,32 SAY aresp_pu
DO valiseto WITH bandlset, reglset
DO CASE
CASE .NOT. bandlset .AND. .NOT. reglset
LOOP
CASE (.NOT. bandlset .AND. reglset) .OR.
(bandlset .AND. .NOT. reglset)
DO leencp
DO valncp WITH bandncp, regncp
DO CASE
CASE .NOT. bandncp .AND. .NOT. regncp
LOOP
CASE (bandncp .AND. .NOT. regncp) .OR.
(.NOT. bandncp .AND. regncp)

```

```

DO i=ocpu
DO valmod WITH bandmod
IF bandmod
DO valcpu WITH bandcpu
IF bandcpu
DO valcomp WITH bandcomp
IF bandcomp
@ 10,54 SAY '#SUC : '
@ 10,80 GET anum_sucur PICTURE '999'
@ 10,84 SAY '#C REG : '
@ 10,72 GET anum_c_reg PICTURE '999'
READ
DO valsucc WITH bandsuc,regsuc
DO CASE
CASE (bandsuc .AND. .NOT. regsuc) ;
.OR. (.NOT. bandsuc .AND. regsuc)

```

```

*****
**** Se leen los datos de la sucursal y se registran ****
**** los datos nuevos. ****
*****

```

```

DO altasuc
DO altamod
@ 24,00 SAY borraestado
@ 24,pietitulo SAY mensajefin
fin = 0

```

```

&& Loop para detectar Esc o
&& Enter.

```

```

DO WHILE fin <> 27 .AND. fin <> 13
fin = INKEY()
IF fin = 27
modi = .F.
ENDIF

```

```

ENDDO
DO inivar
CASE .NOT. bandsuc .AND. .NOT. regsuc
LOOP

```

```

ENDCASE
ELSE
LOOP

```

```

ENDIF
ELSE
LOOP

```

```

ENDIF
ELSE
LOOP

```

```

ENDIF
ENDCASE

```

```

ENDCASE
ENDIF

```

```

ENDIF
ENDIF

```



```

ENDIF
ELSE
  IF LASTKEY() = 27                && Loop para detectar Esc.
    RETURN
  ENDIF
  DO mlune                         && Envia mensaje de error.
ENDIF
ENDDO
RESTSCREEN(04,01,21,78,pantatms)  && Restablece pantalla mp.
SET COLOR TO n/bg,w/n             && Regresa a rutina menuatms
RETURN
PROCEDURE ayudal3
DO ayudamod                       && Invoca rutina de ayuda.

```

```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina   : Conscajo.                               *****
*****
*****      Funcion  : Permitir la consulta de datos de     *****
*****                  cajeros en la opcion de modificar.   *****
*****
*****      Rutina que la invoca : Atasmodi.                 *****
*****
*****      Rutinas que invoca   : Ninguna.                  *****
*****
*****
*****

```

```

*****
*****      Variables.                                       *****
*****
*****

```

```

@ 08,18 SAY 'TIPO :'                                     && Presenta los datos para
@ 08,24 GET atipo_atm PICTURE '!!!!!!'                 && ser modificados.
@ 08,32 SAY 'ENLACE :'
@ 08,40 GET aenlace PICTURE '!!!!!!'
@ 08,51 SAY 'RESPALDO :'
@ 08,61 GET aresp_enla PICTURE '!!!!!!'
@ 09,09 SAY 'PU :'
@ 09,13 GET apu PICTURE '!!!!!!'
@ 09,22 SAY 'RESPALDO :'
@ 09,32 GET aresp_pu PICTURE '!!!!!!'
@ 09,41 SAY 'LSET :'
@ 09,47 GET alineaset PICTURE '!!!!!!'
@ 09,56 SAY 'FECHA LIB :'
@ 09,67 GET afecha_lib
RETURN
&& Regresa a rutina atasmodi

```

```

*****
*****
***** Proyecto : Control de Cajeros Automaticos. *****
***** Rutina : Valpucon. *****
***** Funcion : Validar en la opcion de modificar *****
***** cuando una PU no es aceptada. *****
***** Rutina que la invoca : Atmsmodi. *****
***** Rutina que invoca : Mpu. *****
*****
*****
*****
*****

```

```

*****
***** Variables. *****
*****
*****

```

```

PARAMETER bandpu
bandpu = .T.
SELECT 1 && Busca es cajero.
SET ORDER TO 1
SEEK alu
IF apu <> pu && Valida que la pi no se
SET ORDER TO 2 && duplique.
SEEK apu
IF FOUND()
DO mpu && Envia mensaje de error.
bandpu = .F.
ENDIF
ENDIF
RETURN && Regresa a rutina atmsmodi

```

```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****      Rutina   : Vallseto.                             *****
*****      Funcion  : Validar linesets en la opcion de      *****
*****                  modificar.                            *****
*****      Rutina que la invoca : Atmsmodi.                 *****
*****      Rutinas que invoca   : Misetne.                  *****
*****
*****
*****

```

```

*****
*****      Variables.                                       *****
*****
*****

```

```

PARAMETER bandiset, regiset
bandiset = .T.
regiset = .T.
barraestado = SPACE(80)
@ 24,00 SAY barraestado
SELECT 1
SET ORDER TO 1
SEEK alu
IF lineset = alineset
bandiset = .T.
regiset = .F.
ELSE
SELECT 2
SEEK alineset
IF FOUND()
bandiset = .T.
regiset = .F.
anop = nop
SELECT 3
SEEK anop
amod_nop = str(mod_nop)
acpu = cpu
SELECT 4
SEEK acpu
ao_computo = c_computo
ELSE
bandiset = .F.
DO misetne WITH regiset
if regiset
anop = SPACE(03)
ENDIF
ENDIF
ENDIF
RETURN

```

&& Selecciona el archivo
 && de cajeros.

&& Verifica si el lineset
 && ya existe.

&& Obtiene informacion de
 && nop,cpu,etc.

&& Envía mensaje de confir -
 && macion para registrar el
 && lineset.

&& Regresa a rutina atmsmodi


```

anom_c_reg = nom_c_regl
atel_reg = tel_reg
afax_reg = fax_reg
ELSE
bandsuc = .F.                                && Si es nueva sucursal pide
DO msaucrne WITH regsuc                       && la informacion.
IF regsuc                                     && Envia mensaje para confir-
anom_sucur = SPACE(15)                       && mar el registro de inf.
atot_atms = SPACE(01)                         && Inicializa las variables.
acoordinad = SPACE(30)
ahora_serv = SPACE(15)
atel_suc = SPACE(10)
afax_suc = SPACE(10)
scalle = SPACE(15)
anumero = SPACE(08)
acolonias = SPACE(20)
acp = SPACE(05)
adelegacio = SPACE(15)
aestado = SPACE(15)
apoblacion = SPACE(20)
anom_c_reg = SPACE(20)
atel_reg = SPACE(10)
afax_reg = SPACE(10)
ENDIF
ENDIF
ENDIF
RETURN
&& Regresa a rutina atmsmodi

```



```

SELECT 3
APPEND BLANK
                                && Actualiza informacion de
                                && ncps.

REPLACE nop WITH anop, mod_nop WITH VAL(amod_nop), cpu WITH acpu
SELECT 4
SEEK acpu
IF .NOT. FOUND()
    APPEND BLANK
    REPLACE cpu WITH acpu, c_computo WITH ac_computo
ENDIF

ENDIF
SELECT 6
SEEK anum_sucur + anum_c_reg
IF .NOT. FOUND()
    APPEND BLANK
    REPLACE num_sucurs WITH anum_sucur, num_c_reg WITH anum_c_reg
    REPLACE nom_sucurs WITH anom_sucur, total_atms WITH (total_atms + 1)
    REPLACE coordinado WITH acoordinad, hora_serv WITH ahora_serv
    REPLACE tel_suc WITH atel_suc , fax_suc WITH afax_suc
    REPLACE calle WITH acalle , numero WITH anumero
    REPLACE colonia WITH acolonia , cp WITH VAL(acop)
    REPLACE delegacion WITH adelegacio, poblacion WITH apoblacion

    SELECT 7
    SEEK val(anum_c_reg)
    IF .NOT. FOUND()
        APPEND BLANK
        REPLACE num_c_reg WITH anum_c_reg, nom_c_reg WITH anon_c_reg
        REPLACE tel_reg WITH atel_reg , fax_reg WITH afax_reg
    ENDIF
ELSE
    REPLACE num_sucurs WITH anum_sucur, num_c_reg WITH anum_c_reg
    REPLACE nom_sucurs WITH anom_sucur
    REPLACE coordinado WITH acoordinad, hora_serv WITH ahora_serv
    REPLACE tel_suc WITH atel_suc , fax_suc WITH afax_suc
    REPLACE calle WITH acalle , numero WITH anumero
    REPLACE colonia WITH acolonia , cp WITH VAL(acop)
    REPLACE delegacion WITH adelegacio, poblacion WITH apoblacion
    SELECT 7
    SEEK val(anum_c_reg)
    IF .NOT. FOUND()
        APPEND BLANK
        REPLACE num_c_reg WITH anum_c_reg, nom_c_reg WITH anon_c_reg
        REPLACE tel_reg WITH atel_reg , fax_reg WITH afax_reg
    ENDIF
ENDIF
RETURN
                                && Regresa a rutina atmmodi

```

```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina   : Ayudamod.                               *****
*****
*****      Funcion  : Proporcionar ayuda en la pantalla de   *****
*****                  modificacion de datos de cajeros.     *****
*****
*****      Rutina que la invoca : Atmsmodi.                  *****
*****
*****      Rutina que invoca   : Ninguna.                    *****
*****
*****
*****

```

```

panta = SAVESCREEN(04,02,21,79)      && Salva pantalla atmsmodi.

```

```

SET COLOR TO bg/n

```

```

@ 05,35,21,78 BOX

```

```

SET COLOR TO n/bg,w/n

```

```

@ 04,34 TO 20,77

```

```

@ 05,35 SAY '          Ayuda          1/2 '

```

```

@ 08,35 SAY '

```

```

@ 07,35 SAY ' Para hacer modificaciones a la informa -

```

```

@ 08,35 SAY ' ción presentada, posicione en el campo

```

```

@ 09,35 SAY ' deseado y teclee la nueva información.

```

```

@ 10,35 SAY ' Use las flecha para ir al campo que

```

```

@ 11,35 SAY ' se va a corregir.

```

```

@ 12,35 SAY ' Las características de los campos son :

```

```

@ 13,35 SAY ' LU : 4 dígitos numéricos . Ej. 4950

```

```

@ 14,35 SAY ' TIPO : sólo puede ser DIEBOLD ó IBM.

```

```

@ 15,35 SAY ' ENLACE : radio, lp, lc, microondas, etc.

```

```

@ 16,35 SAY ' RESPALDO : Es el respaldo del enlace y

```

```

@ 17,35 SAY ' puede ser alguno de los valo-

```

```

@ 18,35 SAY ' res para el enlace.

```

```

@ 19,35 SAY ' <PgDw> Más...

```

```

salir = 0

```

```

DO WHILE salir <> 27 .AND, salir <> 3      && Loop para detectar Esc o

```

```

    salir = INKEY()      && PgDw.

```

```

ENDDO

```

```

IF salir = 3

```

```

@ 05,35 SAY '          Ayuda          2/2 '

```

```

@ 08,35 SAY '

```

```

@ 07,35 SAY ' PU : es la unidad física. Ej. PUI59N01

```

```

@ 08,35 SAY ' RESPALDO : es el respaldo de la PU.

```

```

@ 09,35 SAY '          Ej. PUI59R01

```

```

@ 10,35 SAY ' LSET : es el lineset al que se encuentra

```

```

@ 11,35 SAY '          conectada la PU.

```

```

@ 12,35 SAY ' FECHA LIB : fecha de liberación del ATM.

```

```

@ 13,35 SAY ' NCP : ctrl. de comunicaciones. Ej. j80

```

```

@ 14,35 SAY ' MODELO : modelo del NCP. Ej. 3745

```

```

@ 15,35 SAY ' CPU : procesador central. Ej. S90C

```

```

@ 16,35 SAY ' C COMP : centro de cómputo. Ej. CSJ      '
@ 17,35 SAY ' #SUC : sucursal que encargada del ATM.    '
@ 18,35 SAY ' #C REG : estado de la república.         '
@ 18,35 SAY '                                     < Esc > Salir '
ENDIF
salir = 0
DO WHILE salir <> 27                                && Loop para detectar Esc.
  salir = INKEY()
ENDDO
RESTSCREEN(04,02,21,79,pante)
SET COLOR TO n/bg,w/n                               && Regresa a rutina atsmodi
RETURN

```



```
SET COLOR TO bg/b,w/n
@ 04,01 CLEAR TO 21,78
CLOSE ALL
RETURN
```

&& Regresa a rutina mp.

ENDIF

ELSE

escape = 0

ENDIF

SET COLOR TO bg/n

@ 05,48,10,73 BOX ' J-L'

SET COLOR TO n/bg,w/n

@ 04,47 TO 09,72

@ 24,00 SAY borraestado

SET MESSAGE TO 24 CENTER

@ 05,48 PROMPT ' ALTA DE RESOLVEDOR

' MESSAGE 'Registrar datos'+
'nuevos'

@ 08,48 PROMPT ' BAJA DE RESOLVEDOR

' MESSAGE 'Borrar datos'

@ 07,48 PROMPT ' MODIF DE RESOLVEDOR

' MESSAGE 'Modificar datos'+
'erroneas'

@ 08,48 PROMPT ' CONSULTA DE RESOLVEDOR

' MESSAGE 'Consultar' +
'informacion'

MENU TO choice

```
*****
***** Invoca rutina de la opcion seleccionada. *****
*****
```

DO CASE

CASE choice = 1

DO diraltas

escape = 1

SET COLOR TO bg/b,w/n

@ 04,01 CLEAR TO 21,78

&& Invoca rutina para alta
&& de resolvedores.

CASE choice = 2

DO dirbajas

escape = 1

SET COLOR TO bg/b,w/n

@ 04,01 CLEAR TO 20,78

&& Invoca rutina para baja
&& de resolvedores.

CASE choice = 3

DO dirmodif

escape = 1

SET COLOR TO bg/b,w/n

@ 04,01 CLEAR TO 21,78

&& Invoca rutina para modi-
&& ficaciones.

CASE choice = 4

DO dirconsu

escape = 1

SET COLOR TO bg/b,w/n

@ 04,01 CLEAR TO 21,78

&& Invoca rutina para con-
&& sultas.

ENDCASE

ENDDO

CLOSE ALL

RETURN

&& Regresa a rutina mp.

PROCEDURE ayuda40

DO ayudadir

&& Rutina de ayuda para.

```

*****
*****
***** Proyecto : Control de Cajeros Automaticos. *****
***** Rutina : Ayudadir. *****
***** Funcion : Proporcionar ayuda en la pantalla de *****
***** menu para resolvidores. *****
***** Rutina que la invoca : Menudir. *****
***** Rutina que invoca : Ninguna. *****
*****
*****
*****

```

```

pante = SAVESCREEN(04,02,21,79)                && Salva pantalla menudir.
SET COLOR TO bg/n
@ 05,35,21,78 BOX                               ' | 4-L| |'
SET COLOR TO n/bg,w/n
@ 04,34 TO 20,77
@ 05,35 SAY '                                     Ayuda                                     '
@ 06,35 SAY '                                     '
@ 07,35 SAY ' Use las flechas para posicionarse en la '
@ 08,35 SAY ' opción que desea. Al estar en la opción '
@ 09,35 SAY ' deseada, pulse la tecla <Enter> para se- '
@ 10,35 SAY ' leccionarla. '
@ 11,35 SAY ' Use la tecla < F1 > para obtener infor - '
@ 12,35 SAY ' mación de como usar el sistema. '
@ 13,35 SAY ' La tecla <Esc> siempre le permitirá re - '
@ 14,35 SAY ' gresar a la pantalla anterior. '
@ 15,35 SAY ' '
@ 16,35 SAY ' '
@ 17,35 SAY ' '
@ 18,35 SAY ' '
@ 19,35 SAY ' <ESC> Salir '
salir = 0
DO WHILE salir <> 27                             && Loop para detectar Esc.
  salir = INKEY()
ENDDO
RESTSCREEN(04,02,21,79,pante)                   && Restablece pantalla.
SET COLOR TO n/bg,w/n
RETURN                                           && Regresa a rutina menudir.

```



```

DO valnres WITH bandres
IF bandres
    @ 12,20 SAY 'NOMBRE :'
    @ 12,20 GET anom_resol PICTURE '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'
    @ 12,59 SAY 'TEL :'
    @ 12,64 GET atel_parti PICTURE '9999999'
    @ 13,09 SAY 'AREA :'
    @ 13,15 GET aarea PICTURE '!!!!!!!!!!!!'
    @ 13,49 SAY 'EXTENSION :'
    @ 13,60 GET aextension PICTURE '!!!!!!!!!!!!'
    CLEAR GETS

    @ 13,26 SAY 'APLICACION :' && Lee aplicacion.

    @ 13,38 GET aaplicacio PICTURE '!!!!!!!!!!!!'
    READ
    IF LASTKEY () = 27 && Detecta si se ha presio-
        RETURN && nado Esc.
    ENDIF
ELSE
    @ 12,20 SAY 'NOMBRE :' && Lee datos si la clave
    && no existe.

    @ 12,20 GET anom_resol PICTURE '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'
    @ 12,59 SAY 'TEL :'
    @ 12,64 GET atel_parti PICTURE '9999999'
    @ 13,09 SAY 'AREA :'
    @ 13,15 GET aarea PICTURE '!!!!!!!!!!!!'
    @ 13,26 SAY 'APLICACION :'
    @ 13,38 GET aaplicacio PICTURE '!!!!!!!!!!!!'
    @ 13,49 SAY 'EXTENSION :'
    @ 13,60 GET aextension PICTURE '!!!!!!!!!!!!'
    READ
    IF LASTKEY () = 27 && Detecta si se ha presio-
        RETURN && nado Esc.
    ENDIF
ENDIF
DO valnrapi WITH bandnrapi && Valida la clave.
IF bandnrapi
    DO valarea WITH bandarea && Valida el area.
    IF bandarea
        @ 24,00 SAY borraestado
        @ 24,pietitulo2 SAY mensaje2 && Mensaje para validar alta
        valida = 0 && Loop para detectar Enter
        && o Esc.

    DO WHILE valida <> 13 .AND. valida <> 27
        valida = INKEY()
        IF valida = 13
            DO altares && Invoca rutina para regis-
                && trar los datos.
            @ 24,00 SAY borraestado
            @ 24,pietitulo1 SAY mensaje1
            fin = 0

```



```

                                && Loop para detectar Enter
                                && o Esc.
DO while fin <> 27 .AND. fin <> 13
  fin = INKEY()
  IF fin = 27
    alta = .F.
 ENDIF
ENDDO
DO inivardi                                && Inicializa variables.
ENDIF
ENDDO
ELSE
  DO inivardi                                && Inicializa variables.
ENDIF
ENDDO
RESTSCREEN(04,01,21,78,pantdir)            && Restablece pantalla.
SET COLOR TO n/bg,w/n
RETURN                                       && Regresa a rutina menudir.
PROCEDURE ayuda41                           && Invoca rutina de ayuda.
DO ayudaltr

```



```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****      Rutina   : Valires.                               *****
*****      Funcion  : Validar cuando una clave de resolvidor *****
*****                  ha sido de alta previamente.         *****
*****      Rutina que la invoca : Diraltas,dirmodif.        *****
*****      Rutina que invoca   : Ninguna.                   *****
*****
*****
*****
*****

```

```

*****
*****      Variables.      *****
*****
*****

```

```

PARAMETER bandres
bandres= .T.
SELECT 1                                && Selecciona archivo de
SET ORDER TO 1                          && resolvidores.
SEEK anum_resol
IF FOUND()
    anom_resol = nom_resolv
    atel_parti = STR(tel_partic)
    aarea      = area
    aextension = extension
ELSE
    bandres = .F.
ENDIF
RETURN                                  && Regresa a rutina diraltas
                                        && dirmodif.

```

```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****      Rutina   : Valnrapl.                               *****
*****      Funcion  : Validar cuando un resolvidor tiene   *****
*****                  asignada una aplicacion.             *****
*****      Rutina que la invoca : Diraltss.                 *****
*****      Rutina que invoca   : Mrese.                     *****
*****
*****
*****

```

```

*****
*****      Variables.      *****
*****
*****

```

```

PARAMETER bandnrapl
bandnrapl = .T.
SELECT 2                                && Selecciona archivo de
SET ORDER TO 1                          && aplicaciones.
SEEK anum_resol + aaplicacion
IF FOUND()
    DO mrese                             && Envia mensaje de error.
    bandnrapl = .F.
ENDIF
RETURN                                  && Regresa a rutina diraltss

```

```

.....
.....
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina      : Mreso.                               *****
*****
*****      Funcion     : Enviar mensaje cuando un resolvidor *****
*****                   ya tiene asignada una aplicacion.   *****
*****
*****      Rutina que la invoca : Valnrapi.                  *****
*****
*****      Rutina que invoca   : Ninguna.                   *****
*****
.....
.....

```

```

.....
*****      Variables.      *****
.....

```

```

borraestado = SPACE(80)
mensaje      = 'Resolvidor dado de alta anteriormente. < Esc > Continuar'

pistitulo = (79-LEN(mensaje))/2      && Calcula longitud del
@ 24,00 SAY borraestado              && mensaje.
@ 24,pistitulo SAY mensaje
salir = 0
DO WHILE salir <> 27                 && Detecta Esc.
  salir = INKEY()
ENDDO
RETURN                               && Regresa a rutina valnrapi

```

```

.....
.....
****
****      Proyecto : Control de Cajeros Automaticos.      ****
****
****      Rutina   : Valarea.                               ****
****
****      Funcion  : Validar cuando una area no           ****
****                  ha sido de alta previamente.        ****
****
****      Rutina que la invoca : Diraitas,dirmodif.        ****
****
****      Rutina que invoca   : Mareane.                   ****
****
.....

```

```

.....
****      Variables.      ****
.....

```

```

PARAMETER bandarea
bandarea= .T.
SELECT 1
SET ORDER TO 2
SEEK aarea
IF .NOT. FOUND()
    DO mareane WITH bandarea
ENDIF
RETURN

&& Selecciones archivo de
&& resolvedores.

&& Envia mensaje de confir-
&& macion para registrar
&& una area.
&& Regresa a rutine diraitas
&& dirmodif.

```

```

.....
.....
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****      Rutina    : Mareane.                               *****
*****      Funcion   : Enviar mensaje cuando una area no   *****
*****                  ha sido de alta anteriormente.      *****
*****      Rutina que la invoca : Valarea.                   *****
*****      Rutina que invoca   : Ninguna.                    *****
*****
*****
.....

```

```

.....
*****      Variables.      *****
.....

```

```

PARAMETER bandarea
bandarea = .T.
borraestado = SPACE(80)
mensaje = 'Area no registrada.   _ Registrar   <Esc> Corregir'

pietitulo = (79-LEN(mensaje))/2      && Calcula longitud del
@ 24,00 SAY borraestado              && mensaje.
@ 24,pietitulo SAY mensaje

salir = 0
DO WHILE salir <> 27 .AND. salir <> 13  && Loop para detectar Enter
    salir = INKEY()                    && o Esc.
    IF salir = 27
        bandarea = .F.
    ENDIF
ENDDO
RETURN                                && Regresa a rutina valarea.

```

```

.....
.....
.....
Proyecto : Control de Cajeros Automaticos.
.....
Rutina : Altres.
.....
Funcion : Grabar informacion de resolvedores
en archivos.
.....
Rutina que la invoca : Diraltas.
.....
Rutinas que invoca : Ninguna.
.....
.....
.....

```

```

.....
..... Variables.
.....
.....

```

```

SELECT 1                                && Selecciona archivo de
SET ORDER TO 1                          && resolvedores.
SEEK anum_resol
IF .NOT. FOUND()
  APPEND BLANK
  REPLACE num_resolv WITH anum_resol, nom_resolv WITH anam_resol
  REPLACE area WITH aarea , extension WITH aextension
  REPLACE tel_partic WITH VAL(ate1_parti)
endif
IF bandnrap)
  SELECT 2                                && Selecciona archivo de
  APPEND BLANK                            && aplicaciones.

  REPLACE num_resolv WITH anum_resol, aplicacion WITH aaplicacio
ENDIF
RETURN                                    && Regresa a rutina diraltas

```



```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina   : Dirbajas.                               *****
*****
*****      Funcion  : Eliminar datos de resolvedores.       *****
*****
*****      Rutina que la invoca : Menudir.                   *****
*****
*****      Rutinas que invoca   : inivardi,valida,m baja,     *****
*****                          bajares,ayudabar.             *****
*****
*****
*****

```

```

*****
*****      Variables.                                         *****
*****
*****

```

```

fin          = 0
baja         = .T.
bandbaja    = .T.
bandnrapl   = .T.
SET KEY 28 TO ayuda42
mensajefin  = 'Información eliminada.  —  && Asigna F1 como ayuda.
borraestado = SPACE(80)                  Continuar <Esc> Terminar'
pletitulo   = (79-LEN(mensajefin))/2    && Calcula longitud del
pantdir     = SAVESCREEN(04,01,21,78)   && mensaje.
SET COLOR TO bg/n
@ 12,08,16,78 BOX ' |'-L|'
SET COLOR TO n/bg,w/n
@ 11,07 TO 15,77
@ 12,08 CLEAR TO 14,78
@ 24,00 SAY borraestado
DO inivardi
DO WHILE baja
bandbaja    = .T.
bandnrapl   = .T.
@ 12,08 CLEAR TO 14,78
@ 24,00 SAY borraestado
@ 12,09 SAY 'CLAVE :'
&& Lee la clave y la aplica-
&& cion.
@ 12,16 GET anum_resol PICTURE '999'
@ 12,20 SAY 'APLICACION : '
@ 12,32 GET aaplicacio PICTURE '!!!!!!!!!!!!'
READ
IF LASTKEY () = 27
RETURN
&& Detecta Esc.
&& Regresa a rutina menudir.
ENDIF
DO valida WITH bandnrapl
IF bandnrapl
&& Valida si exista la clave
&& Despliega informacion a
&& dar de baja.

```

```

@ 12,43 SAY 'AREA :'
@ 12,49 GET aarea PICTURE '!!!!!!'
@ 13,09 SAY 'NOMBRE :'
@ 13,17 GET anom_resol PICTURE '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'
@ 13,48 SAY 'TEL :'
@ 13,54 GET atel_parti PICTURE '9999999'
@ 13,62 SAY 'EXT :'
@ 13,67 GET aestension PICTURE '!!!!!!'
CLEAR GETS
DO mbaja WITH bandbaja && Envia mensaje de confir-
IF bandbaja && acion para la baja.
DO bajares && Elimina los datos.
@ 24,00 SAY borraestado
@ 24,pistitulo SAY mensajefin
fin = 0
&& Loop para detectar Enter
&& o Esc.
DO WHILE fin <> 27 .AND. fin <> 13
  fin = INKEY()
  IF fin = 27
    baja = .F.
  ENDIF
ENDDO
DO inivardi && Inicializa variables.
ENDIF
ENDIF
ENDDO
RESTSCREEN(04,01,21,78,pantdir) && Restablece pantalla.
SET COLOR TO n/bg,w/n
RETURN && Regresa a rutina menudir.
PROCEDURE ayuda42
DO ayudabar && Invoca rutina de ayuda.

```

```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****      Rutina   : Valida.                               *****
*****
*****      Funcion  : Validar claves de resolvidores       *****
*****                  en la opcion de bajas.             *****
*****
*****      Rutina que la invoca : Dirbajas.                *****
*****
*****      Rutina que invoca   : Mresno.                   *****
*****
*****
*****
*****

```

```

*****
*****      Variables.                                     *****
*****
*****

```

```

PARAMETER bandnrapi
bandnrapi = .T.
SELECT 2
SET ORDER TO 1
SEEK anum_resol + aplicacion
IF .NOT. FOUND()
    DO mresno
    bandnrapi = .F.
ELSE
    SELECT 1
    SET ORDER TO 1
    SEEK anum_resol
    anom_resol = nom_resolv
    aarea      = area
    aextension = extension
    atel_parti = tel_partic
ENDIF
RETURN

```

&& Selecciona archivo de
&& aplicaciones.

&& Envia mensaje de error.

&& Selecciona archivo de
&& resolvidores.

&& Regresa a rutina dirbajas


```

*****
*****
***** Proyecto : Control de Cajeros Automaticos. *****
***** Rutina : Mbaja. *****
***** Funcion : Enviar mensaje para confirmar la *****
***** baja de resolvidores. *****
***** Rutina que la invoca : Dirbajas. *****
***** Rutina que invoca : Ninguna. *****
*****
*****
*****

```

```

*****
***** Variables. *****
*****

```

```

PARAMETER bandbaja
bandbaja = .T.
borraestado = SPACE(80)
mensaje = 'Confirme.      — Eliminar      <Eso> Corregir'

pletitulo = (79-LEN(mensaje))/2      && Calcula longitud del
@ 24,00 SAY borraestado              && mensaje.
@ 24,pletitulo SAY mensaje
salir = 0
DO WHILE salir <> 27 .AND. salir <> 13  && Loop para detectar Enter
    salir = INKEY()                  && o Esc.
    IF salir = 27
        bandbaja = .F.
    ENDIF
ENDDO
RETURN                               && Regresa a rutina dirbajas

```

```

*****
*****
***** Proyecto : Control de Cajeros Automaticos. *****
*****
***** Rutina : Bajares. *****
*****
***** Funcion : Borrar informacion de resolvedores *****
***** en archivos. *****
*****
***** Rutina que la invoca : Dirbajas. *****
*****
***** Rutinas que invoca : Ninguna. *****
*****
*****
*****

```

```

*****
***** Variables. *****
*****
*****

```

```

SELECT 2                               && Selecciona archivo de
SET ORDER TO 1                          && aplicaciones.
SEEK anum_resol + aplicacio
DELETE
PACK
SET ORDER TO 2
SEEK anum_resol
IF .NOT. FOUND()                        && Selecciona archivo de
  SELECT 1                               && resolvedores.
  SET ORDER TO 1
  SEEK anum_resol
  DELETE
  PACK
ENDIF
RETURN                                  && Regresa a rutina dirbajas

```



```

@ 12,20 GET anom_resol PICTURE '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'
READ
IF LASTKEY () = 27          && Detecta si se ha presio-
RETURN                    && nado Esc.
ENDIF
DO CASE
CASE anom_resol =SPACE(03) && Envia mensaje para soli-
DO mdat                   && citar datos.
DO inivardi                && Valida el nombre.
OTHERWISE

DO valnomr WITH bandnomr
IF bandnomr
DO leeres                 && Obtiene los datos del
                        && archivo resolvidores.
DO leeapl                 && Obtiene los datos del
                        && archivo aplicaciones.

CLEAR GETS
@ 24,00 SAY borraestado
@ 24,pietitulo SAY mensajefin
fin = 0
                        && Loop para detectar Enter
                        && o Esc.

DO WHILE fin <> 27 .AND. fin <> 13
    fin = INKEY()
    IF fin = 27
        cons = .F.
    ENDIF
ENDDO
DO inivardi                && Inicializa variables.
ENDIF

ENDCASE
OTHERWISE
DO valnres WITH bandres
IF bandres
DO leeres                 && Obtiene los datos del
                        && archivo resolvidores.
DO leeapl                 && Obtiene los datos del
                        && archivo aplicaciones.

CLEAR GETS
@ 24,00 SAY borraestado
@ 24,pietitulo SAY mensajefin
fin = 0
                        && Loop para detectar Enter
                        && o Esc.

DO WHILE fin <> 27 .AND. fin <> 13
    fin = INKEY()
    IF fin = 27
        cons = .F.
    ENDIF
ENDDO
DO inivardi                && Inicializa variables.
ELSE
DO mresne                  && Envia mensaje de error.
ENDIF

```

ENDCASE
ENDDO
RESTSCREEN(04,01,21,70,pantdir)
SET COLOR TO n/bg,w/n
RETURN

PROCEDURE ayuda44
DO ayudacor

&& Restablece pantalla.

&& Regresa a rutina dirconsu

&& Invoca rutina de ayuda.


```

.....
****
****      Proyecto : Control de Cajeros Automaticos.      ****
****
****      Rutina   : Leeapl.                                ****
****
****      Funcion  : Presentar datos de las aplicaciones   ****
****                  manejadas por los resolvedores.     ****
****
****      Rutina que la invoca : Dirconsu.                 ****
****
****      Rutinas que invoca   : Ninguna.                  ****
****
.....

```

```

.....
****      Variables.      ****
.....

```

```

renglon = 13
ocolumna = 47
SELECT 2                                && Selecciona archivo de
GO TOP                                  && aplicaciones.
LOCATE FOR num_resolv = anum_resol
DO WHILE .NOT. EOF()
  @aplicacio = aplicacion
  @ renglon,columna SAY 'APLICACION : '  && Presenta los datos.
  columna = columna + 12
  @ renglon,columna GET @aplicacio
  columna = columna + 11
  IF columna > 65                        && Cambia de renglon.
    renglon = renglon + 1
    columna = 09
  ENDIF
CONTINUE
ENDDO
RETURN                                  && Regresa a rutina dirconsu

```

```

.....
*****
****
****      Proyecto : Control de Cajeros Automaticos.      ****
****      Rutina   : Valnres.                               ****
****      Funcion  : Validar cuando una clave de resolvedor ****
****                ha sido de alta previamente.          ****
****      Rutina que la invoca : Diraitas,dirmodif.        ****
****      Rutina que invoca   : Ninguna.                   ****
****
.....

```

```

.....
****                               Variables.                               ****
.....

```

```

PARAMETER bandnres
bandnres= .T.
SELECT 1                               && Selecciona archivo de
SET ORDER TO 1                          && resolvedores.
SEEK anum_resol
IF FOUND()
    anum_resol = nom_resolv
    atel_part1 = STR(tel_partic)
    aarea      = area
    aextension = extension
ELSE
    bandnres = .F.
ENDIF
RETURN                                  && Regresa a rutina diraitas
                                         && dirmodif.

```



```

aplica(5) = SPACE(10)
registro(1) = 0
registro(2) = 0
registro(3) = 0
registro(4) = 0
registro(5) = 0
@ 12,08 CLEAR TO 14,78
@ 24,00 SAY borraestado
@ 12,09 SAY 'CLAVE :'
@ 12,18 GET anum_resol PICTURE '999' && Lee la clave.
READ
IF LASTKEY ( ) = 27 && Detecta si se ha presio-
RETURN && nado Esc.
ENDIF
DO CASE
CASE anum_resol = SPACE(03) && Si no se proporciona la
@ 12,20 SAY 'NOMBRE :' && clave se pide el nombre.
@ 12,28 GET anum_resol PICTURE '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'
READ
IF LASTKEY ( ) = 27 && Detecta si se ha presio-
RETURN && nado Esc.
ENDIF
DO CASE
CASE anum_resol =SPACE(03) && Envia mensaje para pedir
DO mdata && los datos.
DO inivardi
OTHERWISE && Valida el nombre.

DO valnomr WITH bandnomr
IF bandnomr
DO leeresm && Presenta los datos para
&& ser modificados.
READ
IF LASTKEY ( ) = 27 && Detecta si se ha presio-
RETURN && nado Esc.
ENDIF && Valida el area.

DO valarea WITH bandarea
IF bandarea
DO leerapl && Valida la aplicacion.

DO valapl WITH bandapl
IF bandapl && Registre las modifica-
&& ciones.
DO regmodif
@ 24,00 SAY borraestado
@ 24,pietitulo SAY mensaje:fin
fin = 0
&& Loop para detectar Enter
&& o Esc.

DO WHILE fin <> 27 .AND. fin <> 13
fin = INKEY()
309

```

```

IF fin = 27
    modi = .F.
ENDIF
ENDDO
DO inivardi      && Inicializa variables.
ENDIF
ENDIF
ENDCASE
OTHERWISE
DO valnes WITH bandres      && Valida la clave.
IF bandres
DO leereem      && Presenta los datos para
READ          && ser modificados.
IF LASTKEY () = 27      && Detecta Esc.
RETURN
ENDIF
DO valarea WITH bandarea    && Valida el area.
IF bandarea
DO leaplml
DO valapl WITH bandapl      && Valida la aplicacion.
IF bandapl
DO regmodif      && Registra las modifica-
&& ciones.
@ 24.00 SAY borraestado
@ 24.pietitulo SAY mensajefin
fin = 0
&& Loop para detectar Enter
&& o Esc.
DO while fin <> 27 .AND. fin <> 13
    fin = INKEY()
    IF fin = 27.
        modi = .F.
    ENDIF
ENDDO
DO inivardi      && Inicializa variables.
ENDIF
ELSE
DO mreene      && Envia mensaje de error.
ENDIF
ENDCASE
ENDDO
RESTSCREEN(04,01,21,78,pantdir)      && Restablece pantalla.
SET COLOR TO n/bg,w/n
RETURN      && regresa a rutina menudir.
PROCEDURE ayuda43      && Invoca rutina de ayuda.
DO ayudamor

```



```

*****
*****
***** Proyecto ; Control de Cajeros Automaticos. *****
***** Rutina ; Regmodif. *****
***** Funcion ; Grabar informacion de resolvidores *****
***** en archivos. *****
***** Rutina que la invoca ; Dirmodif. *****
***** Rutinas que invoca ; Ninguna. *****
*****
*****
*****

```

```

*****
***** Variables. *****
*****

```

```

x = 1                               && indice para los arreglos.
SELECT 1                             && Selecciona archivo de
SET ORDER TO 1                       && resolvidores.
SEEK anum_resol
  REPLACE nom_resolv WITH anum_resol
  REPLACE area WITH aarea, extension WITH aextension
  REPLACE tel_partic WITH VAL(atel_parti)
SELECT 2                               && Selecciona archivo de
DO WHILE registro[x] <> 0              && aplicaciones.
  GO registro[x]
  REPLACE aplicacion WITH aplica[x]
  x = x + 1
ENDDO
RETURN                               && Regresa a rutina dirmodif

```



```

*****
*****
****
****      Proyecto : Control de Cajeros Automaticos.      ****
****
****      Rutina   : Ayudamor.                             ****
****
****      Funcion  : Proporcionar ayuda en la pantalla de  ****
****                  modificaciones de resolvedores.      ****
****
****      Rutina que la invoca : Dirmodif.                 ****
****
****      Rutina que invoca   : Ninguna.                   ****
****
*****
*****

```

```

pante = SAVESCREEN(04,02,21,79)      && Salva pantalla dirmodif.
SET COLOR TO bg/n
@ 05,35,21,78 BOX                    ' [ ] - L [ ] '
SET COLOR TO n/bg,w/n
@ 04,34 TO 20,77
@ 05,35 SAY '                               Ayuda                               1/1 '
@ 06,35 SAY '
@ 07,35 SAY ' Las modificaciones se realizan de dos '
@ 08,35 SAY ' formas diferentes: por CLAVE o NOMBRE. '
@ 09,35 SAY ' Al ser solicitado alguno de los campos '
@ 10,35 SAY ' anteriores, proporcione su valor o [ ] '
@ 11,35 SAY ' para pasar al siguiente campo. '
@ 12,35 SAY ' Las características de los campos son: '
@ 13,35 SAY ' CLAVE : tres digitos numéricos . Ej. 015 '
@ 14,35 SAY ' NOMBRE : Campo alfabético. '
@ 15,35 SAY ' TEL : Campo de siete digitos. Ej 7400285 '
@ 16,35 SAY ' AREA : Campo de hasta diez caracteres. '
@ 17,35 SAY ' EXTENSION : Campo numérico '
@ 18,35 SAY ' APLICACION : Campo de diez caracteres. '
@ 19,35 SAY '                               < Esc > '
salir = 0
DO WHILE salir <> 27                  && Loop para detectar Esc.
    salir = INKEY()
ENDDO
RESTSCREEN(04,02,21,79,pante)        && Restablece pantalla.
SET COLOR TO n/bg,w/n
RETURN                                && Regresa a rutina dirmodif

```


111.5.5 MODULO REPORTES.

```

*****
*****
***** Proyecto : Control de Cajeros Automaticos. *****
***** Rutina : Menurepo. *****
***** Funcion : Presentar opciones para reportes *****
***** de cajeros y directorio. *****
***** Rutina que la invoca : MP. *****
***** Rutina que invoca : Menuratms, menurres, *****
***** ayudarep. *****
*****
*****
*****

```

```

*****
***** Variables. *****
*****
*****

```

```

borraestado = SPACE(80)
escape = 0
SET KEY 26 TO ayuda30 && Asigna F1 como ayuda.

```

```

*****
***** Imprime menu para reportes. *****
*****

```

```

SET PATH TO c:\tesis\reportes
SET WRAP ON
MENU TO choice
DO WHILE choice <> 3
  IF escape = 0 && Detecta si se ha presio-
    escape = LASTKEY() && nado Esc.
    IF escape = 27
      SET COLOR TO bg/b,w/n
      @ 04,01 CLEAR TO 21,78
      RETURN
    ENDIF
  ELSE
    escape = 0
  ENDIF
  SET COLOR TO bg/n
  @ 05,35,08,55 BOX ' |'-'|'
  SET COLOR TO n/bg,w/n
  @ 04,35 TO 07,54

```

MENU TO choice

```
.....  
***** Invoca rutina de la opcion seleccionada. *****  
.....
```

DO CASE

CASE choice = 1

DO menuratm

escape = 1

SET COLOR TO bg/b,w/n

@ 04,01 CLEAR TO 21,78

CASE choice = 2

DO menurres

escape = 1

SET COLOR TO bg/b,w/n

@ 04,01 CLEAR TO 20,78

ENDCASE

ENDDO

RETURN

PROCEDURE ayuda30

DO ayudarep

&& invoca rutina para menu
&& de reportes de cajeros.

&& invoca rutina para menu
&& de reportes de directorio

&& Regresa a rutina mp.

&& invoca rutina de ayuda.

```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina   : Ayudarep.                             *****
*****
*****      Funcion  : Proporcionar ayuda en la pantalla de  *****
*****                  menu para reportes.                 *****
*****
*****      Rutina que la invoca : Menurepo.                 *****
*****
*****      Rutina que invoca   : Ninguna.                   *****
*****
*****
*****

```

```

pante = SAVESCREEN(04,02,21,79)      && Salva pantalla menurepo.
SET COLOR TO bg/n
@ 05,35,21,78 BOX
SET COLOR TO n/bg,w/n
@ 04,34 TO 20,77
@ 05,35 SAY '                               Ayuda                               '
@ 08,35 SAY '
@ 07,35 SAY ' Use las flechas para posicionarse en la '
@ 08,35 SAY ' opción que desee. Al estar en la opción '
@ 09,35 SAY ' deseada, pulse la tecla <Enter> para se '
@ 10,35 SAY ' leccionaria.
@ 11,35 SAY ' Use la tecla < F1 > para obtener infor - '
@ 12,35 SAY ' mación de como usar el sistema.
@ 13,35 SAY ' La tecla <Esc> siempre le permitirá re - '
@ 14,35 SAY ' gresar a la pantalla anterior.
@ 15,35 SAY '
@ 16,35 SAY '
@ 17,35 SAY '
@ 18,35 SAY '
@ 19,35 SAY '
salir = 0
DD WHILE salir <> 27      && Loop para detectar Esc.
    salir = INKEY()
ENDDD
RESTSCREEN(04,02,21,79,pante)      && Restablece pantalla.
SET COLOR TO n/bg,w/n
RETURN      && Regresa a rutina menurepo

```



```

anumero      = SPACE(08)
acolonias   = SPACE(20)
acp          = SPACE(05)
adelegacio  = SPACE(15)
aestado     = SPACE(15)
apoblacion  = SPACE(20)
anom_c_reg  = SPACE(20)
atel_reg    = SPACE(10)
afax_reg    = SPACE(10)

```

```

&& Variables del archivo
&& regionales.

```

```

SELECT 2 USE cajeros INDEX caj_lu, caj_pu

```

```

&& Se abre areas para los
&& archivos.

```

```

SELECT 3 USE linesets INDEX lset_lset

```

```

SELECT 4 USE ncps INDEX ncp_ncp

```

```

SELECT 5 USE cpus INDEX cpu_cpu

```

```

SELECT 6 USE regional INDEX reg_numr

```

```

SELECT 7 USE enlaces INDEX enl_enl

```

```

&& Asigna F1 como ayuda.

```

```

SET KEY 28 TO ayuda31

```

```

*****
***** Imprime menu para reportes de cajeros. *****
*****

```

```

pantratm = SAVESCREEN(04,01,21,78)

```

```

&& Salva pantalla menurepo.

```

```

SET WRAP ON

```

```

MENU TO choice

```

```

DO WHILE choice <> 7

```

```

  escape = LASTKEY()

```

```

  IF escape = 27

```

```

    SET COLOR TO bg/b,w/n

```

```

    @ 04,01 CLEAR TO 21,78

```

```

    CLOSE ALL

```

```

    RETURN

```

```

  ENDIF

```

```

  SET COLOR TO bg/n

```

```

  @ 06,55,12,77 BOX ' | - L | '

```

```

  SET COLOR TO n/bg,w/n

```

```

  @ 05,54 TO 11,78

```

```

  @ 24,00 SAY borraestado

```

```

  SET MESSAGE TO 24 center

```

```

  @ 06,55 PROMPT 'POR NUM DE SUCURSAL ' MESSAGE 'Listado por # de' + ;
  'sucursal'

```

```

  @ 07,55 PROMPT 'POR NOM DE SUCURSAL ' MESSAGE 'Listado por nombre' + ;
  'de sucursal'

```

```

  @ 08,55 PROMPT 'POR CENTRO REGIONAL ' MESSAGE 'Listado por centro' + ;
  'regional'

```

```

  @ 09,55 PROMPT 'POR NCP ' MESSAGE 'Listado por NCP'

```

```

  @ 10,55 PROMPT 'LISTA DE SUCURSALES ' MESSAGE 'Listado de sucursales'

```

```

  MENU TO choice

```

```

*****
***** Invoca rutina de23a opcion seleccionada. *****
*****

```

```

DO CASE
CASE choice = 1
DO ratmnum
    && invoca reporte por numero
    && de sucursal.
CASE choice = 2
DO ratanoms
    && invoca reporte por nombre
    && de sucursal.
CASE choice = 3
DO ratmnumr
    && invoca reporte por numero
    && de centro regional.
CASE choice = 4
DO rncp
    && invoca reporte por ncp.
CASE choice = 5
DO raucs
    && invoca reporte de datos
    && de sucursal.
ENDCASE

ENDDO
RESTSCREEN(04,01,21,78,pantratm)
    && Restablece pantalla
    && menurepo.
SET COLOR TO n/bg,w/n
CLOSE ALL
    && Regresa a rutina menurepo
RETURN
    && invoca rutina de ayuda.

PROCEDURE ayuda31
DO ayudratm

```



```

renglon = 08
columna = 01
SELECT 1
GO TOP
DO WHILE num_c_regi = '001'
  anum_sucur = num_sucurs
  anum_c_reg = num_c_regi
  anom_sucurs = nom_sucurs
  atot_atms = total_atms
  SELECT 2
  GO TOP
  LOCATE FOR (num_sucurs = anum_sucur .AND. num_c_regi = anum_c_reg)
  atipo_atm = tipo_atm
  aenlace = enlace
  alineset = lineset
  SELECT 3
  SEEK alineset
  ancp = ncp
  @ renglon,columna SAY anum_sucurs + SPACE(01) + anom_sucurs
  @ renglon,21 SAY 'ENLACE:' + aenlace + SPACE(01) + 'LSET: ' +
    alineset
  @ renglon,56 SAY 'NCP: ' + ancp + SPACE(03) + 'ATMS: ' + atipo_atm
  renglon = renglon + 2
  SELECT 2
  GO TOP
  LOCATE FOR (num_sucurs = anum_sucur .AND. num_c_regi = anum_c_reg)
  DO WHILE .NOT. EOF()
    alu = lu
    apu = pu
    @ renglon,columna SAY STR(alu) + SPACE(01) + apu
    renglon = renglon + 1
    IF renglon >= 60
      @ renglon,34 SAY pagina
      pagina = pagina + 1
      @ 01,25 SAY 'CENTRO DE COMPUTO SAN JUAN.'
      @ 02,15 SAY 'LISTADO DE CAJEROS POR NUMERO DE SUCURSAL.'
      @ 03,32 SAY DTOC( DATE() )
      @ 05,01 SAY '-----'+
        renglon = 06
      @ renglon,columna SAY anum_sucurs + SPACE(01) + anom_sucurs
      @ renglon,21 SAY 'ENLACE: ' + aenlace + SPACE(01) + 'LSET: ' +
        alineset
      @ renglon,56 SAY 'NCP: ' + ancp + SPACE(03) + 'ATMS: ' +
        atipo_atm
      renglon = renglon + 2
    ENDIF
    CONTINUE
  ENDDO
  renglon = renglon - 1
  @ renglon,64 SAY 'TOTAL: ' + STR(atot_atms) + SPACE(01) + 'CAJEROS'
  renglon = renglon + 1
  @ renglon,01 SAY '-----'+
    renglon = renglon + 1

```

```
SELECT 1
SKIP
ENDDO
EJECT
SET DEVICE TO SCREEN
ELSE
  @ 24,00 SAY borraestado
  @ 24,pietitulo2 SAY mensaje2
  salir = 0
  DO WHILE salir <> 13
    salir = INKEY()
  ENDDO
ENDIF
SET COLOR TO n/bg,w/n
RETURN
```

```
&& Selecciona archivo de
&& sucursales.
&& Termina la impresion
```

```
&& mensaje de error si la
&& impresora no esta lista.
&& Loop para detectar Esc.
```

```
&& Regresa a rutina menurstm
```

```

*****
*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina   : Ratmnoms.                               *****
*****
*****      Funcion  : Imprimir listado de cajeros          *****
*****                  por nombre de sucursal.              *****
*****
*****      Rutina que la invoca : Menuratm.                 *****
*****
*****      Rutina que invoca   : Valimp,inivar.             *****
*****
*****
*****
*****

```

```

*****
*****      Variables.      *****
*****
*****

```

```

SELECT 1                                && Selecciona archivo de su-
      use suc_noms                        && cursales ordenado por
                                          && nombre.
                                          && Inicializa variables.

DO inivar
bandimp   = .T.
borraestado = SPACE(80)
mensaje1  = 'Prepare la impresora.      ── Continuar'
mensaje2  = 'Impresora no esta lista.   ── Continuar'
mensaje3  = 'Imprimiendo...'
pietitulo1 = (79-LEN(mensaje1))/2      && Calcula longitud de los
pietitulo2 = (79-LEN(mensaje2))/2      && mensajes.
pietitulo3 = (79-LEN(mensaje3))/2
renglón   = 0
columna   = 0
pagina    = 1
@ 24,00 say borraestado
@ 24,pietitulo1 SAY mensaje1
salir = 0
DO WHILE salir <> 13                    && Loop para detectar Enter.
  salir = INKEY()

ENDDO

DO valimp WITH bandimp                  && Valida si la impresora
IF bandimp                              && esta lista.
  @ 24,00 SAY borraestado
  @ 24,pietitulo3 SAY mensaje3
  SET DEVICE TO PRINTER                 && Imprime listado.

  @ 01,25 SAY 'CENTRO DE COMPUTO SAN JUAN.'
  @ 02,15 SAY 'LISTADO DE CAJEROS POR NOMBRE DE SUCURSAL.'
  @ 03,32 SAY dtoc(date())
  @ 05,01 SAY '-----';
  renglón = 05

```

```

columna = 01
SELECT 1
GO TOP
DO WHILE num_c_regi = '001'
  anum_sucur = num_sucurs
  anum_c_reg = num_c_regi
  anom_sucur = nom_sucurs
  atot_atms = total_atms
  SELECT 2
  GO TOP
  LOCATE FOR (num_sucurs = anum_sucur .AND. num_c_regi = anum_c_reg)
  atipo_atm = tipo_atm
  aenlace = enlace
  alineset = lineset
  SELECT 3
  SEEK alineset
  ancp = ncp
  @ renglon,columna SAY anom_sucurs + SPACE(01) + anum_sucurs
  @ renglon,21 SAY 'ENLACE:' + aenlace + SPACE(01) + 'LSET:' + ;
  alineset
  @ renglon,56 SAY 'NCP:' + ancp + SPACE(03) + 'ATMS:' + atipo_atm
  renglon = renglon + 2
  SELECT 2
  GO TOP
  LOCATE FOR (num_sucurs = anum_sucur .AND. num_c_regi = anum_c_reg)
  DO WHILE .NOT. EOF()
    alu = lu
    apu = pu
    @ renglon,columna SAY STR(alu) + SPACE(01) + apu
    renglon = renglon + 1
    IF renglon >= 80
      @ renglon,34 SAY pagina
      pagina = pagina + 1
      @ 01,25 SAY 'CENTRO DE COMPUTO SAN JUAN.'
      @ 02,15 SAY 'LISTADO DE CAJEROS POR NOMBRE DE SUCURSAL.'
      @ 03,32 SAY DTOC(DATE())
      @ 05,01 SAY '-----'+;
      renglon = 08
      @ renglon,columna SAY anom_sucurs + SPACE(01) + anum_sucurs
      @ renglon,21 SAY 'ENLACE:' + aenlace + SPACE(01)+ 'LSET:' ;
      + alineset
      @ renglon,56 SAY 'NCP:' + ancp + SPACE(03) + 'ATMS:' + ;
      atipo_atm
      renglon = renglon + 2
    ENDIF
  CONTINUE
ENDDO
renglon = renglon - 1
@ renglon,64 SAY 'TOTAL:' + STR(atot_atms) + SPACE(01) + 'CAJEROS'
renglon = renglon + 1
@ renglon,01 SAY '-----'+;
renglon = renglon + 1

```

```

SELECT 1
SKIP
ENDDO
EJECT
SET DEVICE TO SCREEN
ELSE
@ 24,00 SAY borraestado
@ 24,pietitulo2 SAY mensaje2
salir = 0
DO WHILE salir <> 13
    salir = INKEY()
ENDDO
ENDIF
SET COLOR TO n/bg,w/n
RETURN

```

```

&& Selecciona archivo de
&& sucursales.

```

```

&& Envia mensaje de error.

```

```

&& Loop para detectar Esc.

```

```

&& Regresa a rutina menuratm

```



```

.....
.....
.....
Proyecto : Control de Cajeros Automaticos.
.....
Rutina : Ratmnuar.
.....
Funcion : Imprimir listado de cajeros
por numero de centro regional.
.....
Rutina que la invoca : Menurata.
.....
Rutina que invoca : Valimp,inivar.
.....
.....

```

```

.....
.....
Variables.
.....

```

```

SELECT 1                               && Selecciona archivo de
      USE suc_nums                       && sucursales.

DO inivar
bandimp = .T.
borraestado = SPACE(80)
mensaje1 = 'Prepare la impresora.      — Continuar'
mensaje2 = 'Impresora no esta lista.   — Continuar'
mensaje3 = 'Imprimiendo...'
pletitulo1 = (79-LEN(mensaje1))/2      && Calcula longitud de los
pletitulo2 = (79-LEN(mensaje2))/2      && mensajes.
pletitulo3 = (79-LEN(mensaje3))/2
renghon = 0
columna = 0
pagina = 1
regional = '000'
@ 24,00 SAY borraestado
@ 24,pletitulo1 SAY mensaje1
salir = 0
DO WHILE salir <> 13                    && Loop para detectar Enter.
  salir = INKEY()
ENDDO
DO valimp WITH bandimp                  && Valida si la impresora
IF bandimp                              && esta lista.
@ 24,00 SAY borraestado
@ 24,pletitulo3 SAY mensaje3
SET DEVICE TO PRINTER                  && Imprime listado.

@ 01,25 SAY 'CENTRO DE COMPUTO SAN JUAN.'
@ 02,18 SAY 'LISTADO DE CAJEROS POR CENTRO REGIONAL.'
@ 03,32 SAY DTOC(DATE())
@ 05,01 SAY '-----'
renghon = 08

```

```

columna = 01
SELECT 1
GO TOP
DO WHILE .NOT. EOF()
  anum_sucur = num_sucurs
  anum_c_reg = num_c_regi
  anom_sucur = nom_sucurs
  atot_atms = total_atms
  SELECT 2
  GO TOP
  LOCATE FOR (num_sucurs = anum_sucur .AND. num_c_regi = anum_c_reg)
  atipo_atm = tipo_atm
  aenlace = enlace
  alineset = lineset
  SELECT 3
  SEEK alineset
  ancp = ncp
  SELECT 4
  LOCATE FOR num_c_regi = anum_c_reg
  anom_c_reg = nom_c_regi
  IF regional <> anum_c_reg
    @ renglon,17 SAY 'CENTRO REGIONAL: ' + anum_c_reg + SPACE(02) ;
    + anom_c_reg
    renglon = renglon + 1
    @ renglon,01 SAY '-----'+;
    '-----'
    renglon = renglon + 1
    regional = anum_c_reg
  ENDIF
  @ renglon,columna SAY anum_sucurs + SPACE(01) + anom_sucurs
  @ renglon,21 SAY 'ENLACE: ' + aenlace + SPACE(01)+ 'LSET: ' + ;
  alineset
  @ renglon,56 SAY 'NCP: ' + ancp + SPACE(03) + 'ATMS: ' + atipo_atm
  renglon = renglon + 2
  SELECT 2
  GO TOP
  LOCATE FOR (num_sucurs = anum_sucur .AND. num_c_regi = anum_c_reg)
  DO WHILE .NOT. EOF()
    alu = lu
    apu = pu
    @ renglon,columna SAY STR(alu) + SPACE(01) + apu
    renglon = renglon + 1
    IF renglon >= 60
      @ renglon,34 SAY pagina
      pagina = pagina + 1
      @ 01,25 SAY 'CENTRO DE COMPUTO SAN JUAN.'
      @ 02,16 SAY 'LISTADO DE CAJEROS POR CENTRO REGIONAL.'
      @ 03,32 SAY DTOC(DATE())
      @ 05,01 SAY '-----'+;
      '-----'
      renglon = 06
      @ renglon,columna SAY anum_sucurs + SPACE(01) + anom_sucurs
      @ renglon,21 SAY 'ENLACE: ' + aenlace + SPACE(01)+ 'LSET: ' ;
      + alineset
    
```

```

      @ renglon,56 SAY 'NCP: ' + ancp + SPACE(03) + 'ATMS: ' + ;
      atipo_atm
      renglon = renglon + 2
    ENDIF
    CONTINUE
  ENDDO
  renglon = renglon - 1
  @ renglon,84 SAY 'TOTAL: ' + STR(atot_atms) +SPACE(01) + 'CAJEROS'
  renglon = renglon + 1
  @ renglon,01 SAY '-----';
  renglon = renglon + 1
  SELECT 1
  SKIP
  ENDDO
  EJECT
  SET DEVICE TO SCREEN
ELSE
  @ 24,00 SAY borraestado
  @ 24,petititulo2 SAY mensaje2
  salir = 0
  DO WHILE salir <> 13
  salir = INKEY()
  ENDDO
ENDIF
SET COLOR TO n/bg,w/n
RETURN

```

&& Selecciona archivo de
 && sucursales.

&& Envia mensaje de error.

&& Loop para detectar Enter.

&& Regresa a rutina menuram


```

@ 02,31 SAY 'LISTADO DE CAJEROS POR NCP.'
@ 03,40 SAY DTOC(DATE())
@ 05,01 SAY '-----'+
-----+
renghon = 06
columna = 01
SELECT 1                               && Selecciona archivo de
GO TOP                                  && linesets.
DO WHILE (.NOT. EOF())
  alineset = lineset
  ancp     = ncp
  IF ancp <> ancp
    aancp = ancp
    @ renghon,38 SAY 'NCP : ' + ancp
    renghon = renghon + 1
    @ renghon,01 SAY '-----'+
-----+
    renghon = renghon + 1
  ENDIF
SELECT 2                               && Selecciona archivo de
GO TOP                                  && cajeros.
LOCATE FOR (lineset = alineset)
DO WHILE (lineset = alineset)
  aiu     = lu
  anum_sucur = num_sucurs
  anum_c_reg = num_c_reg
  atipo_atm = tipo_atm
  aenlace  = enlace
  apu      = pu
  SELECT 3                               && Selecciona archivo de
                                          && sucursales.

LOCATE FOR (num_sucurs = anum_sucur .AND.
            num_c_reg1 = anum_c_reg)
anom_sucur = nom_sucurs
atot_atms = total_atms
regional = anum_c_reg
@ renghon,columna SAY anum_sucurs + SPACE(01) + anom_sucurs
@ renghon,21 SAY 'ENLACE: ' + aenlace + SPACE(01)+ 'LSET: ' +
alineset
@ renghon,56 SAY 'CR: ' + anum_c_reg + SPACE(03) + 'ATMS: ' +
atipo_atm
renghon = renghon + 2
SELECT 2                               && Selecciona archivo de
GO TOP                                  && cajeros.

LOCATE FOR (num_sucurs=anum_sucur .AND. num_c_reg1= anum_c_reg)
DO WHILE (num_sucurs=anum_sucur .AND. num_c_reg1= anum_c_reg)
  @ renghon,columna SAY STR(aiu) + SPACE(01) + apu
  renghon = renghon + 1
  IF renghon >= 60                       && Cambia de pagina.
    @ renghon,34 SAY pagina
    pagina = pagina + 1
    @ 01,31 SAY 'CENTRO DE COMPUTO SAN JUAN.'
    @ 02,31 SAY 'LISTADO DE CAJEROS POR NCP.'
    @ 03,40 SAY DTOC(DATE())

```

```

@ 05,01 SAY '-----';
renqlon = 06
@ renqlon,columna SAY anum_sucurs + SPACE(01) +
      anum_sucurs
@ renqlon,21 SAY 'ENLACE: ' + anenlace + SPACE(01) +
      'LSET: ' + alineaset
@ renqlon,56 SAY 'CR: ' + anum_c_reg + SPACE(03) +
      'ATMS: ' + atipo_atm
renqlon = renqlon + 2
ENDIF
SKIP
ENDDO
ENDDO
renqlon = renqlon - 1
@ renqlon,64 SAY 'TOTAL: ' + STR(atot_atms) + SPACE(01) +
      'CAJEROS'
renqlon = renqlon + 1
@ renqlon,01 SAY '-----';
renqlon = renqlon + 1
SELECT 1
SKIP
ENDDO
EJECT
SET DEVICE TO SCREEN
ELSE
@ 24,00 SAY borraestado
@ 24,pletitulo2 SAY mensaje2
salir = 0
DO WHILE salir <> 13
salir = INKEY()
ENDDO
ENDIF
SET COLOR TO n/bg,w/n
RETURN

```

&& Selecciona archivo de
&& lineaseta.

&& Envia mensaje de error.

&& Loop para detectar Enter.

&& Regresa a rutina menuradm


```

columna = 01
SELECT 1
GO TOP
DO WHILE .NOT. EOF()
  anum_sucur = num_sucurs
  anum_c_reg = num_c_regi
  anom_sucur = nom_sucurs
  atot_atms = total_atms
  acoordinad = coordinado
  ahora_serv = hora_serv
  atel_suc = tel_suc
  afax_suc = fax_suc
  acalle = calle
  annumero = numero
  acolonia = colonia
  acp = str(cp)
  a delegacio = delegacion
  aestado = estado
  apoblacion = poblacion
  SELECT 0
  LOCATE for num_c_regi = anum_c_reg && Selecciona archivo de
  anom_c_reg = nom_c_regi && centros regionales.
  IF regional (<) anum_c_reg
    @ renglon,17 SAY 'CENTRO REGIONAL: ' + anum_c_reg + SPACE(02) ;
    + anom_c_reg
    renglon = renglon + 1
    @ renglon,01 SAY '-----';
    '-----';
    renglon = renglon + 1
    regional = anum_c_reg
  ENDIF
  @ renglon,25 SAY 'SUCURSAL: ' + anum_sucurs + SPACE(05) + ;
  anom_sucurs
  renglon = renglon + 2
  @ renglon,columna SAY 'HORARIO DE SERVICIO: ' + ahora_serv + ;
  SPACE(04) + 'COORDINADOR: ' + acoordinad
  renglon = renglon + 1
  @ renglon,columna SAY 'TELEFONO: ' + atel_suc + SPACE(10) + ;
  'FAX: ' + afax_suc
  renglon = renglon + 1
  @ renglon,columna SAY 'CALLE: ' + acalle + SPACE(08) + 'NUM: ' + ;
  annumero + SPACE(05) + 'COL: ' + acolonia
  renglon = renglon + 1
  @ renglon,columna SAY 'CP: ' + acp + SPACE(21) + 'DEL: ' + ;
  a delegacio
  renglon = renglon + 1
  @ renglon,columna SAY 'EDO: ' + aestado + SPACE(10) + ;
  'POBLACION: ' + apoblacion
  renglon = renglon + 1
  @ renglon,columna SAY '-----';
  '-----';
  renglon = renglon + 1
  IF renglon >= 55 && Cambia de pagina.
    @ renglon+1,34 SAY pagina
    pagina = pagina + 1
    @ 01,25 SAY 'CENTRO DE COMPUTO SAN JUAN.'
  338

```



```

@ 02,20 SAY 'LISTADO DE SUCURSALES.'
@ 03,32 SAY DTOC(DATE())
@ 05,01 SAY '-----'+;
'-----'
renqlon = 08
ENDIF
SELECT 1                                && Selecciona archivo de
SKIP                                    && sucursales.
ENDDO
EJECT
SET DEVICE TO SCREEN
ELSE
@ 24,00 SAY borraestado
@ 24,p1ettitulo2 SAY mensaje2          && Envia mensaje de error.
salir = 0
DO WHILE salir <> 13                   && Loop para detectar Enter.
    salir = INKEY()
ENDDO
ENDIF
SET COLOR TO n/bg,w/n
RETURN                                  && Regresa a rutina menurata

```

```

.....
.....
****
****      Proyecto : Control de Cajeros Automaticos.      ****
****
****      Rutina   : Vallap.                                ****
****
****      Funcion  : Validar cuando la impresora no       ****
****                  esta lista.                          ****
****
****      Rutina que la invoca : Ratmnuus,ratmnoms,        ****
****                               ratmnumr,rnop,rucos,      ****
****                               rresnom,rresarea,rresapl.  ****
****
****      Rutina que invoca   : Ninguna.                  ****
****
.....
.....

```

```

.....
****
****      Variables.                                     ****
****
.....

```

```

PARAMETER bandimp
bandimp = .T.
bandimp = ISPRINTER()

RETURN

&& Detecta si la impresora
&& esta lista.
&& Regresa a rutina ratmnume
&& ratmnoms,ratmnumr,rnop,
&& rucos,rresnom,rresarea,
&& rresapl.

```

```

*****
*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina   : Ayudratm.                             *****
*****
*****      Funcion  : Proporcionar ayuda en la pantalla de   *****
*****                  menu para reportes de cajeros.       *****
*****
*****      Rutina que la invoca : Menurاتم.                 *****
*****
*****      Rutina que invoca   : Ninguna.                   *****
*****
*****
*****

```

```

pante = SAVESCREEN(04,02,21,79)      && Salva pantalla-menurاتم.
SET COLOR TO bg/n
@ 05,35,21,78 BOX ' | | - | | | '
SET COLOR TO n/bg,w/n
@ 04,34 TO 20,77
@ 05,35 SAY '
@ 06,35 SAY '
@ 07,35 SAY ' Use las flechas para posicionarse en la
@ 08,35 SAY ' opción que desea. Al estar en la opción
@ 09,35 SAY ' deseada, pulso la tecla <Enter> para se-
@ 10,35 SAY ' leccionarla.
@ 11,35 SAY ' Use la tecla < F1 > para obtener infor -
@ 12,35 SAY ' mación de como usar el sistema.
@ 13,35 SAY ' La tecla <Esc> siempre le permitirá re -
@ 14,35 SAY ' gresar a la pantalla anterior.
@ 15,35 SAY '
@ 16,35 SAY '
@ 17,35 SAY '
@ 18,35 SAY '
@ 19,35 SAY '
salir = 0
DO WHILE salir <> 27      && Loop para detectar-Esc.
  salir = INKEY()
ENDDO
RESTSCREEN(04,02,21,79,pante)      && Restablece pantalla.
SET COLOR TO n/bg,w/n
RETURN      && Regresa a ruina menurاتم

```

REPORTES DEL DIRECTORIO.

```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****      Rutina   : Menurres.                            *****
*****      Funcion  : Presentar opciones para reportes     *****
*****                  de resolvedores.                    *****
*****      Rutina que la invoca : Menurepo.                 *****
*****      Rutina que invoca   : Rresnom,rresarea,rresapl,   *****
*****                          ayudarres.                  *****
*****
*****
*****

```

```

*****
*****      Variables.      *****
*****
*****

```

```

borraestado = SPACE(80)
escape      = 0
anum_resol  = SPACE(03)
anom_resol  = SPACE(30)
aextension  = SPACE(09)
aaplicacio  = SPACE(10)
ateparti    = SPACE(07)
aarea       = SPACE(10)

```

```

&& Variables del archivo
&& resolvedores.

```

```

SELECT 1
      USE resolved INDEX res_numr, res_area
      && Se abre areas para los
      && archivos.

```

```

SELECT 2
      USE aplicaci INDEX apl_nrap, apl_nres

```

```

SELECT 3
      USE resarea

```

```

SELECT 4
      USE apl_apl

```

```

SET KEY 26 TO ayuda32
      && Asigna F1 como ayuda.

```

```

*****
*****      Imprime menu para reportes de resolvedores.      *****
*****
*****

```

```

pantratm = SAVESCREEN(04,01,21,78)
SET WRAP ON
      && Salva pantalla menurepo.

```

```

MENU TO choice
DO WHILE choice <> 4
  escape = LASTKEY()
  IF escape = 27
    SET COLOR TO bg/b,w/n
    @ 04,01 CLEAR TO 21,78
    CLOSE ALL
    RETURN
  ENDIF
  SET COLOR TO bg/n
  @ 08,55,10,72 BOX ' |'-'|'!!!'
  SET COLOR TO n/bg,w/n
  @ 05,54 TO 09,71
  @ 24,00 SAY borraestado
  SET MESSAGE TO 24 CENTER
  @ 06,55 PROMPT ' POR NOMBRE ' MESSAGE 'Listado por nombre'
  @ 07,55 PROMPT ' POR AREA ' MESSAGE 'Listado por área'
  @ 08,55 PROMPT ' POR APLICACION ' MESSAGE 'Listado por aplicación'
MENU TO choice

```

```

*****
*****      invoca rutina de la opcion seleccionada.      *****
*****

```

```

DO CASE
CASE choice = 1
  DO rresnom
  SET COLOR TO bg/b,w/n
CASE choice = 2
  DO rresarea
  SET COLOR TO bg/b,w/n
CASE choice = 3
  DO rresapl
  SET COLOR TO bg/b,w/n
ENDCASE

ENDDO
RESTSCREEN(04,01,21,78,pantratm)
SET COLOR TO n/bg,w/n
CLOSE ALL
RETURN

PROCEDURE ayuda32
DO ayudrres

```



```

.....
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****      Rutina   : Rresnom.                               *****
*****      Funcion  : Imprimir listado de resolvidores     *****
*****                  por nombre.                          *****
*****
*****      Rutina que la invoca : Menurres.                 *****
*****
*****      Rutina que invoca   : Inivarre, valimp.          *****
*****
.....

```

```

.....
*****      Variables.                                       *****
.....

```

```

DO inivarre                                     && Inicializa variables.
bandimp = .T.
borraestado = SPACE(80)
mensaje1 = 'Prepare la impresora.      — Continuar'
mensaje2 = 'Impresora no esta lista.   — Continuar'
mensaje3 = 'Imprimiendo...'
p1titulo1 = (79-LEN(mensaje1))/2        && Calcula longitud de los
p1titulo2 = (79-LEN(mensaje2))/2        && mensajes.
p1titulo3 = (79-LEN(mensaje3))/2
@ 24,00 SAY borraestado
@ 24,p1titulo1 SAY mensaje1
salir = 0
DO WHILE salir <> 13                          && Loop para detectar Enter.
  salir = INKEY()
ENDDO
DO valimp WITH bandimp                        && Valida si la impresora
IF bandimp                                   && esta lista.
  @ 24,00 SAY borraestado
  @ 24,p1titulo3 SAY mensaje3
  SET DEVICE TO PRINTER                     && Imprime listado.

  @ 01,25 SAY 'CENTRO DE COMPUTO SAN JUAN.'
  @ 02,17 SAY 'DIRECTORIO DE RESOLVEDORES DE SEGUNDO NIVEL.'
  @ 03,23 SAY 'LISTADO POR NOMBRE,' + SPACE(05) + DTOC( DATE())
  @ 05,01 SAY '-----'+;
  @ 06,05 SAY 'NOMBRE' + SPACE(24) + 'AREA' + SPACE(06) + 'EXTENSION' +;
  @ 07,01 SAY 'SPACE(01) + 'TELEFONO' + SPACE(01) + 'APLICACION'
  @ 07,01 SAY '-----'+;
  renglon = 8
  columna = 2

```

```

SELECT 1
GO TOP
DO WHILE .NOT. EOF()
    anum_resol = num_resolv
    anom_resol = nom_resolv
    aarea = area
    aextension = extension
    atel_parti = str(tel_partic)
    SELECT 2
    GO TOP
    LOCATE FOR num_resolv = anum_resol
    @ renglon,columna SAY anom_resolv +SPACE(01) + aarea + SPACE(02)+
        aextension + SPACE(02) + atel_partic
    DO WHILE .NOT. EOF()
        aplicacio = aplicacion
        @ renglon,65 SAY aplicacio
        renglon = renglon + 1
        IF renglon >= 80
            && Cambia de pagina.

            @ 01,25 SAY 'CENTRO DE COMPUTO SAN JUAN,'
            @ 02,17 SAY 'DIRECTORIO DE RESOLVEDORES DE SEGUNDO NIVEL.'
            @ 03,23 SAY 'LISTADO POR NOMBRE.' + SPACE(05) + DTOC(DATE())
            @ 05,01 SAY '-----';
            @ 06,05 SAY 'NOMBRE' + SPACE(24) + 'AREA' + SPACE(08) + ;
                'EXTENSION' + SPACE(01) + 'TELEFONO' + ;
                SPACE(01) + 'APLICACION'
            renglon = 08
        ENDIF
        CONTINUE
    ENDDO
    @ renglon,01 SAY '-----';
    renglon = renglon + 1
    SELECT 1
    SKIP
    ENDDO
    EJECT
    SET DEVICE TO SCREEN
ELSE
    @ 24,00 SAY borraestado
    @ 24,p1etitulo2 SAY mensaje2
    salir = 0
    DO WHILE salir <> 13
        salir = INKEY()
    ENDDO
    ENDIF
    SET COLOR TO n/bg,w/n
    RETURN
    && Regresa a rutina menurres

```



```

*****
*****
*****      Proyecto : Control de Cajeros Automaticos.      *****
*****
*****      Rutina   : Resarea.                               *****
*****
*****      Funcion  : Imprimir listado de resolvedores     *****
*****                  por area.                             *****
*****
*****      Rutina que la invoca : Menurres.                 *****
*****
*****      Rutina que invoca   : Inivarre, valimp.          *****
*****
*****
*****

```

```

*****
*****      Variables.      *****
*****
*****

```

```

DO inivarre                                && Inicializa variables.
bandimp  = .T.
temporal = SPACE(10)
borraestado = SPACE(80)
mensaje1 = 'Prepare la impresora.      — Continuar'
mensaje2 = 'Impresora no esta lista.  — Continuar'
mensaje3 = 'Imprimiendo...'
pietitulo1 = (79-LEN(mensaje1))/2      && Calcula longitud de los
pietitulo2 = (79-LEN(mensaje2))/2      && mensajes.
pietitulo3 = (79-LEN(mensaje3))/2
@ 24,00 SAY borraestado
@ 24,pietitulo1 SAY mensaje1
salir = 0
DO WHILE salir <> 13                    && Loop para detectar Enter.
  salir = INKEY()
ENDDO
DO valimp WITH bandimp                  && Valida si la impresora
IF bandimp                              && esta lista.
  @ 24,00 SAY borraestado
  @ 24,pietitulo3 SAY mensaje3
  SET DEVICE TO PRINTER                && Imprime listado.

  @ 01,25 SAY 'CENTRO DE COMPUTO SAN JUAN.'
  @ 02,17 SAY 'DIRECTORIO DE RESOLVEDORES DE SEGUNDO NIVEL.'
  @ 03,23 SAY 'LISTADO POR AREA.' + SPACE(07) + DTCO( DATE())
  @ 05,01 SAY '-----'+
  '-----'
  @ 06,02 SAY 'AREA' + SPACE(07) + 'NOMBRE' + SPACE(28) + 'EXTENSION'+
  SPACE(02) + 'TELEFONO' + SPACE(01) + 'APLICACION'
  @ 07,01 SAY '-----'+
  '-----'

  renglon = @
  columna = 2

```

```

SELECT 3
GO TOP
DO WHILE .NOT. EOF()
    anum_resol = num_resolv
    anom_resol = nom_resolv
    aarea = area
    aextension = extension
    atel_parti = str(tel_partic)
    IF temporal <> aarea
        temporal = aarea
    ELSE
        aarea = SPACE(10)
    ENDIF
SELECT 2
GO TOP
LOCATE FOR num_resolv = anum_resol
@ renglon,columna SAY aarea + SPACE(01) + anom_resol + SPACE(02)+;
aextension + SPACE(02) + atel_partic

aarea = temporal
DO WHILE .NOT. EOF()
    aaplicacio = aplicacion
    @ renglon,85 SAY aaplicacio
    renglon = renglon + 1
    IF renglon >= 80
        @ 01,25 SAY 'CENTRO DE COMPUTO SAN JUAN.'
        @ 02,17 SAY 'DIRECTORIO DE RESOLVEDORES DE SEGUNDO NIVEL.'
        @ 03,23 SAY 'LISTADO POR AREA.' + SPACE(07) + DTOC(DATE())
        @ 05,01 SAY '-----';
        @ 06,02 SAY 'AREA' + SPACE(07) + 'NOMBRE' + SPACE(26) + ;
        'EXTENSION' + SPACE(02) + 'TELEFONO' + ;
        SPACE(01) + 'APLICACION'
        renglon = 08
    ENDIF
CONTINUE
ENDDO
SELECT 3
SKIP
IF aarea <> area
    @ renglon,01 SAY '-----';
    renglon = renglon + 1
ENDIF
ENDDO
EJECT
SET DEVICE TO SCREEN
ELSE
@ 24,00 SAY borraestado
@ 24,pietitulo2 SAY mensaje2
salir = 0
DO WHILE salir <> 13
    salir = INKEY()
ENDDO
ENDIF
SET COLOR TO n/bg,w/n
RETURN

```

&& Selecciona archivo de
&& resolvedores ordenado
&& por area.

&& Selecciona archivo de
&& aplicaciones.

&& Cambia de pagina.

&& Selecciona archivo de
&& resolvedores.

&& Envia mensaje de error.

&& Loop para detectar Enter.

&& Regresa a rutina menurres


```

columna = 2
SELECT 4
GO TOP
DO WHILE .NOT. EOF()
  anum_resol = num_resolv
  aplicacio = aplicacion
  IF temporal <> aplicacio
    temporal = aplicacio
  ELSE
    aplicacio = SPACE(10)
  ENDIF
  SELECT 1
  SEEK anum_resol
  anom_resol = nom_resolv
  aarea = area
  aextension = extension
  atel_parti = str(tel_partic)
  @ renglon,columna SAY aplicacion + SPACE(01) + anom_resol +
    SPACE(02) + aextension + SPACE(02) + atel_partic
  @ renglon,85 SAY aarea
  aplicacio = temporal
  renglon = renglon + 1
  IF renglon >= 80
    && Cambia de pagina.

    @ 01,25 SAY 'CENTRO DE COMPUTO SAN JUAN.'
    @ 02,17 SAY 'DIRECTORIO DE RESOLVEDORES DE SEGUNDO NIVEL.'
    @ 03,22 SAY 'LISTADO POR APLICACION.' + SPACE(02) + DTOC(DATE())
    @ 05,01 SAY '-----'+
      '-----'+
      '-----'+
    @ 06,02 SAY 'APLICACION' + SPACE(01) + 'NOMBRE' + SPACE(26) +
      'EXTENSION' + SPACE(02)+ 'TELEFONO'+ SPACE(01) + 'AREA'
    renglon = 08
  ENDIF
  SELECT 4
  SKIP
  IF aplicacio <> aplicacion
    @ renglon,01 SAY '-----'+
      '-----'+
    renglon = renglon + 1
  ENDIF
ENDDO
EJECT
SET DEVICE TO SCREEN
ELSE
  @ 24,00 SAY borraestado
  @ 24,pietitulo2 SAY mensaje2
  salir = 0
  DO WHILE salir <> 13
    salir = INKEY()
  ENDDO
ENDIF
SET COLOR TO n/bg,w/n
RETURN
&& Regresa a rutina menurres

```

III.5.6 MODULO UTILERIA.

```
*****  
*****  
***** Proyecto : Control de Cajeros Automaticos. *****  
***** Rutina : Indexa. *****  
***** Funcion : Crear los indices para las BD. *****  
***** Rutina que la invoca : Mp. *****  
***** Rutinas que invoca : Ninguna. *****  
*****  
*****
```

```
*****  
***** So indexan las Bases de Datos. *****  
*****
```

```
USE cajeros  
INDEX ON lu TO caj_lu  
INDEX ON pu TO caj_pu  
USE linesets  
INDEX ON lineset TO lset_lset  
USE nops  
INDEX ON ncp TO ncp_ncp  
USE cpus  
INDEX ON cpu TO cpu_cpu  
USE tmpo_caj  
INDEX ON lu TO tmpoc_lu  
USE suoursal  
INDEX ON num_sucurs + num_c_regi TO sucsucre  
SORT TO suc_noms ON num_c_regi, num_sucurs  
SORT TO suc_noms ON num_c_regi, num_sucurs  
USE regional  
INDEX ON num_c_regi TO reg_numr  
USE resolved  
INDEX ON num_resolv TO res_numr  
INDEX ON area TO res_area  
SORT ON area TO resarea  
USE aplicaci  
INDEX ON num_resolv + aplicacion TO apl_nrap  
INDEX ON num_resolv TO apl_nres  
SORT ON aplicacion TO apl_apl  
USE enlaces  
INDEX ON enlace TO enl_enl  
CLOSE ALL  
RETURN
```

III.6

MANUAL
DE
USUARIO

MANUAL DE USUARIO PARA EL SISTEMA DE CONTROL DE CAJEROS AUTOMATICOS.

El presente manual tiene como finalidad servir de útil herramienta para resolver alguna duda respecto al manejo del sistema de control de cajeros automáticos.

A lo largo de este manual se describen las características del sistema, así como el funcionamiento de cada una de sus partes. El sistema está dividido en cinco secciones principales:

- Cajeros
- Configuraciones
- Reportes
- Directorio
- Utilerías

Para facilitar el manejo del sistema se utilizan menús con los cuales nos podemos mover de una manera simple dentro del sistema, además de la ayuda permanente con la que se cuenta en cualquier parte del sistema: sólo se necesita oprimir la tecla F1. Aunque deberemos mencionar el uso y la importancia de la tecla ESC, importante para la salida en cualquier parte del sistema si así se deseara.

REQUERIMIENTOS DEL SISTEMA

Se requiere una unidad con las siguientes características:

- Computadora IBM PC-XT o AT o 100 % compatible o de la familia PS/2.
- Unidad de disco flexible de 5 1/4 " ó 3 1/2".
- Puerto paralelo.
- Impresora de 10" ó 15" compatible con IBM.
- 640 kb de memoria RAM.
- Sistema operativo MS-DOS de versión superior a 3.00.

CONVENCIONES SINTACTICAS.

A: B: C:	Se refiere a la unidad de disco .
DRIVE	Unidad de disco.
Enter	Se refiere a la tecla ENTER o RETURN.
Esc	Se refiere a la tecla ESC o escape.
Fn	Se refiere a las teclas de función.

COMO UTILIZAR UN MENU.

La forma más fácil de utilizar un menú es moviéndose con las flechas (derecha, izquierda, arriba, abajo) hasta colocarse en la opción que se desea y después oprimir la tecla Enter. Otra forma de hacerlo es presionando la tecla correspondiente a la primera letra de la opción, lo cual produce el mismo efecto que el método anterior, es decir, moverse con las flechas y pulsar Enter.

Si se quiere pasar de una opción de un submenú al menú principal se puede hacerlo presionando la tecla ESC. Por ejemplo, en el caso del menú anterior si se quisiera cambiar de la opción de CONFIGURACIONES al menú principal, únicamente se deberá presionar la tecla ESC y nos llevará al menú principal sin ningún contratiempo.

USO DEL SISTEMA DE AYUDA.

La tecla F1 permite al usuario contar con un sistema de ayuda que le resuelva las dudas que se le puedan presentar durante el manejo del sistema. Esta ayuda funciona de la siguiente manera:

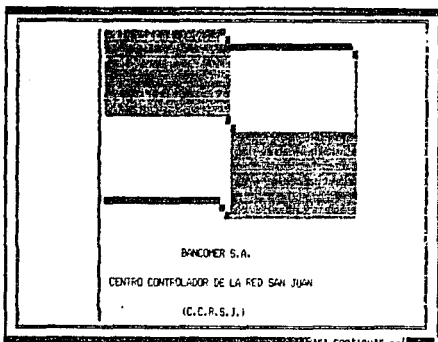
En la pantalla principal del sistema aparece en la esquina inferior derecha el mensaje: F1 AYUDA, de modo que, si en algún momento usted no sabe que hacer deberá oprimir la tecla F1 y aparecerá una ventana que le dará información relacionada con la opción en la que se encuentra actualmente.

La idea de crear este sistema de ayuda fue con el fin de no perder tiempo consultando el manual de usuario cuando posiblemente se trate de una duda sencilla que no requiere de una explicación tan extensa como la que se proporciona en el manual.

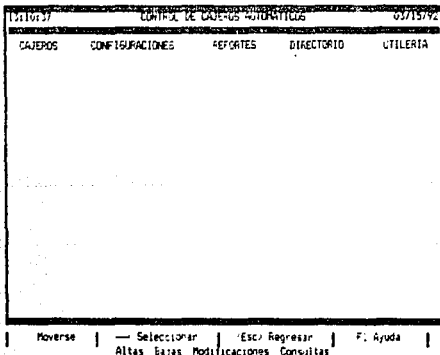
INICIO DE OPERACIONES.

Para entrar al sistema se deberá teclear lo siguiente:

ATMS Enter, con lo que aparece el logo del sistema.



Al presionar Enter el sistema lo lleva al menú principal.



Como se puede observar, en la primera línea se tiene la hora y la fecha actual así como el nombre del sistema. En la siguiente línea se tienen las opciones que conforman el menú principal. La opción que aparece en video inverso es la que puede ser seleccionada al presionar Enter. En la penúltima línea de esta pantalla se especifican las teclas que pueden ser utilizadas durante la sesión de trabajo: Las flechas se usan para moverse de una opción a otra, Enter para seleccionar la opción en la que estemos posicionados, Esc siempre se usa para regresar al nivel anterior y F1 proporciona el sistema de ayuda. En la última línea siempre aparece una pequeña descripción de la opción que esté en video inverso.

A continuación se describe como funciona cada una de las opciones del menú principal.

CAJEROS.

Al estar en la opción CAJEROS y pulsar Enter se abre un submenú que presenta las operaciones que se pueden realizar con la información de cajeros.

13110137		CONTROL DE CAJEROS AUTOMATICOS		03/15/92	
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA	
ALTA DE CAJEROS BAJA DE CAJEROS MODIFICACION CAJEROS CONSULTA DE CAJEROS					

| Moverse | ← Seleccionar | (Esc) Regresar | F1 Ayuda |
Registro de nuevos cajeros

Usando las flechas nos podemos mover a las distintas opciones y al pulsar Enter se selecciona alguna de ellas. F1 proporciona ayuda acerca de lo que se puede hacer en este nivel. Si se pulsa Esc regresamos al nivel anterior (menú principal).

ALTA DE CAJEROS

En esta opción se realiza el registro de la información de nuevos cajeros automáticos que son instalados en la red.

Al entrar a esta opción aparece la siguiente pantalla.

```
13:10:37          CONTROL DE CAJEROS AUTOMATICOS          03/15/92
-----
CAJEROS  CONFIGURACIONES  REPORTES  DIRECTORIO  UTILERIA
-----
ALTA DE CAJEROS
BAJA DE CAJEROS
-----
LU : TIPO : ENLACE : RESPALDO :
PU : RESPALDO : LSET : FECHA LIB #03/15/92
-----
| Moverse | -- Seleccionar | (Esc) Regresar | F1 Ayuda |
```

Aquí se debe proporcionar la siguiente información:

- | | |
|----------|--|
| Lu | Se refiere al número que identifica al cajero, el cual consta de cuatro dígitos numéricos. |
| Tipo | Debe ser Diebold o IBM. |
| Enlace | Es el medio de comunicación (LP, LF, radio, etc). |
| Respaldo | Es el respaldo del enlace y puede ser alguno de los valores para el enlace. |
| Pu | Es la unidad física. Ej. PUI59N01 |

En esta parte se debe proporcionar la siguiente información:

Núm. suc.	Es el número de la sucursal y no puede ser modificado.
Suc.	Nombre de la sucursal.
Horario	Horario de servicio de la sucursal.
Coordinador	Encargado del servicio de cajeros en la sucursal.
Teléfono	Número telefónico de la sucursal.
Fax	Número de fax de la sucursal.
Calle, Núm.	Dirección de la sucursal.
Col. Del.	
C.P.	
Población	Area geográfica de la sucursal.
C. Reg.	Estado de la república.
Teléfono	Número telefónico del centro regional.
Fax	Número Fax del centro regional.

Si todos los datos se dieron correctamente se despliega un mensaje para confirmar el registro de la información, en caso contrario se regresa a realizar las correcciones.

BAJA DE CAJEROS.

En esta opción se realiza la baja de información de cajeros automáticos que son eliminados de la red.

Al entrar a esta opción aparece la siguiente pantalla.

15/10/97	CENTRO DE CAJEROS AUTOMÁTICOS			03/15/97								
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA								
<table border="1"> <tr> <td>ALTA DE CAJEROS</td> <td></td> </tr> <tr> <td>BAJA DE CAJEROS</td> <td></td> </tr> <tr> <td>LU :</td> <td><input type="text"/></td> </tr> <tr> <td></td> <td><input type="text"/></td> </tr> </table>					ALTA DE CAJEROS		BAJA DE CAJEROS		LU :	<input type="text"/>		<input type="text"/>
ALTA DE CAJEROS												
BAJA DE CAJEROS												
LU :	<input type="text"/>											
	<input type="text"/>											
	Moverse		← Seleccionar									
			Esc/ Regresar									
			F1 Ayuda									

El sistema solicita el número de cajero que será eliminado. Si el dato no es correcto se envía un mensaje de error. Si el dato es correcto se despliega la información relacionada y un mensaje para confirmar el borrado de la información, como se muestra.

13:10:37	CONTROL DE CAJEROS AUTOMATICOS			07/15/92																		
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA																		
<table border="1"> <tr> <td>ALTA DE CAJEROS</td> <td></td> </tr> <tr> <td>BAJA DE CAJEROS</td> <td></td> </tr> <tr> <td colspan="2"> LU 14950 TIPO 1 DIEBOLD ENLACE 1 RAD10 RESPALDO 1LP PU 1PU15950 RESPALDO 1PU15950 LSET 1L11050 FEDWA LIB 102/08/92 SUC 1SAM COSME C REG 1MPEA METROPOLITANA </td> </tr> </table>					ALTA DE CAJEROS		BAJA DE CAJEROS		LU 14950 TIPO 1 DIEBOLD ENLACE 1 RAD10 RESPALDO 1LP PU 1PU15950 RESPALDO 1PU15950 LSET 1L11050 FEDWA LIB 102/08/92 SUC 1SAM COSME C REG 1MPEA METROPOLITANA													
ALTA DE CAJEROS																						
BAJA DE CAJEROS																						
LU 14950 TIPO 1 DIEBOLD ENLACE 1 RAD10 RESPALDO 1LP PU 1PU15950 RESPALDO 1PU15950 LSET 1L11050 FEDWA LIB 102/08/92 SUC 1SAM COSME C REG 1MPEA METROPOLITANA																						
<table border="0"> <tr> <td> </td> <td>Moverse</td> <td> </td> <td>— Seleccionar</td> <td> </td> <td>(Esc) Regresar</td> <td> </td> <td>Fl Ayuda</td> <td> </td> </tr> <tr> <td> </td> <td></td> <td> </td> <td>Confirmar</td> <td> </td> <td>— Eliminar</td> <td> </td> <td>(Esc) Corregir</td> <td> </td> </tr> </table>						Moverse		— Seleccionar		(Esc) Regresar		Fl Ayuda					Confirmar		— Eliminar		(Esc) Corregir	
	Moverse		— Seleccionar		(Esc) Regresar		Fl Ayuda															
			Confirmar		— Eliminar		(Esc) Corregir															

MODIFICACION DE CAJEROS

En esta opción se realiza la modificación de información de cajeros automáticos registrados en la red.

Al entrar a esta opción aparece la siguiente pantalla.

13135706	CONTROL DE CAJEROS AUTOMATICOS	03/15/92								
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA						
<table border="1"> <tr> <td>ALTA DE CAJEROS</td> <td></td> </tr> <tr> <td>BAJA DE CAJEROS</td> <td></td> </tr> <tr> <td>LU :</td> <td></td> </tr> </table>					ALTA DE CAJEROS		BAJA DE CAJEROS		LU :	
ALTA DE CAJEROS										
BAJA DE CAJEROS										
LU :										
<table border="1"> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </table>										
	Moverse		— Seleccionar		(Esc) Regresar		F1 Ayuda			

El sistema solicita el número de cajero. Si el dato no es correcto se envía un mensaje de error. Si el dato es correcto se presenta la información para ser modificada.

CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA
ALTA DE CAJEROS BAJA DE CAJEROS				
LU : 0420 TIPO : DIBEROLD ENLACE : RADIO RESPALDO : L.F. PU : PUIBAMCO RESPALDO : PUIBARZO LSET : LINO40 FECHA LIB : 03/15/92 NCP : 1360 MODELO : 3745 CPU : 59% C COMP : 05% #SUC : 040 #C REG : 001				
#SUC : 040 SUC : MDCTEJUMA HORARIO : 08:30AM-15:00PM COORDINADOR : NICHY CONTRERAS TEL : 7-60-34-56 FAX : 621489450 CALLE : AUNTAMIENTO NUM : 105 COL : CENTRO DEL : GUAYATEMOC CP : 14300 POBLACION : DISTRITO FEDERAL C REG : AREA METROPOLITANA TEL : 5340034 FAX :				
Reverse	Selecionar	(Esc) Regresar	F1 Ayuda	
	Confirae.	Registrar	(Esc) Corregir	

Después de realizar las modificaciones se envía un mensaje para registrar las correcciones.

CONSULTA DE CAJEROS

En esta opción se realiza la consulta de información de cajeros automáticos. Al entrar a esta opción aparece la siguiente pantalla.

13/13/96	CONTROL DE CAJEROS AUTOMÁTICOS	03/15/92						
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA				
<table border="1"> <tr> <td>ALTA DE CAJEROS</td> <td></td> </tr> <tr> <td>BAJA DE CAJEROS</td> <td></td> </tr> </table>					ALTA DE CAJEROS		BAJA DE CAJEROS	
ALTA DE CAJEROS								
BAJA DE CAJEROS								
<table border="1"> <tr> <td>LU :</td> <td></td> </tr> </table>					LU :			
LU :								
<table border="1"> <tr> <td></td> <td></td> </tr> </table>								
	Moverse		Seleccionar		Escr Regresar		Fl Ayuda	

Para realizar una consulta es necesario proporcionar el número de cajero. Si el dato no es correcto se envía un mensaje de error. Si el dato es correcto se presenta la información correspondiente al cajero, como se muestra en la siguiente pantalla.

CONFIGURACIONES.

Al entrar a esta opción del menú principal se invoca a un paquete para gráficos en el que se realiza el diseño de las configuraciones de enlace de los cajeros automáticos. La pantalla que aparece es la siguiente.


PROSOFT'S

FANTASY[®]

Version 2.07
06/14/86

Copyright © 1986, W.K. Mason



Press RETURN  to start
ESC for warm restart
Alt-F1 for HELP

Al entrar al área de trabajo el paquete se encuentra en modo texto, siendo posible trabajar en otros tres tipos de modalidades. Estas modalidades son :

- Modo texto.
- Modo dibujo.
- Modo bloque.
- Modo comando.

Los comandos y teclas usadas para trabajar en estos modos son los siguientes.

MODO TEXTO.

El cursor para este modo es una línea vertical y se dispone de los siguientes comandos y teclas especiales.

TECLA O COMANDO.	FUNCION.
Alt-C	Centrar línea.
Alt-D	Borrar línea.
Inicio	Mover el cursor al inicio de la línea.
Fin	Mover el cursor al fin de la línea.
AVPág	Avanzar 1/3 de la pantalla.
RePág	Regresar 1/3 de la pantalla.
Ctl-Inicio	Regresar al inicio del texto.
Ctl-Fin	Avanzar al fin del texto.
Ctl-F1	Seleccionar tipo de letra.

MODO DIBUJO.

En esta modalidad se pueden realizar dibujos con líneas, rectángulos y círculos, además de disponer de comandos y teclas especiales para manipularlos. Este modo se selecciona al presionar la tecla de función F7 y al presionarla nuevamente se regresa a modo texto. El cursor es un punto, al que se puede modificar el tamaño para realizar los dibujos.

TECLA O COMANDO.	FUNCION.
F9	Comenzar a dibujar/Terminar de dibujar.
Flechas	Desplazar el cursor.
F10	Borrar/no borrar.
Alt-L	Dibujar líneas.
F9	Después de seleccionar Alt-L, fija el final de la línea.
Alt-R	Dibujar rectángulos.
Alt-O	Dibujar círculos.
Alt-F	Llenar áreas.
F2	Cancelar la última operación.
F4	Multiplicar por 10 o por 100 el valor de desplazamiento del cursor.
F5	Aumentar el ancho del cursor.
Alt-F5	Disminuir el ancho del cursor.
F6	Aumentar altura del cursor.
Alt-F6	Disminuir la altura del cursor.

Inicio	Movimiento diagonal arriba-izquierda.
Fin	Movimiento diagonal abajo-izquierda.
AvPág	Movimiento diagonal arriba-derecha.
RePág	Movimiento diagonal abajo-derecha.

MODO BLOQUE.

Este modo permite trabajar con áreas parciales o bloques de la pantalla que necesiten algún tratamiento especial.

Los comandos y teclas disponibles son las siguientes.

TECLA O COMANDO.	FUNCION.
Alt-B	Selecciona/inhibir el inicio del bloque.
Flechas	Moverse al fin del bloque.
F9	Mover el bloque.
F10	Copiar el bloque.
R	Rotar el bloque.
D	Borrar el bloque.

MODO COMANDO.

Este modo es usado para trabajar con comandos en lugar de trabajar con combinaciones de teclas y se selecciona al pulsar la tecla Esc.

TECLA O COMANDO.

FUNCION.

SA

Almacenar el archivo de trabajo en disco.

LO

Cargar de disco a memoria un nuevo archivo.

PR

Imprimir archivo.

QU

Terminar la sesión y regresar al menú principal del sistema de cajeros.

FUNCIONES GLOBALES.

Independientemente del modo en que se esté trabajando, existen algunas funciones globales.

TECLA O COMANDO.

FUNCION.

F1

Mostrar la configuración para la sesión actual de trabajo.

F3

Ayuda global para el paquete.

REPORTES.

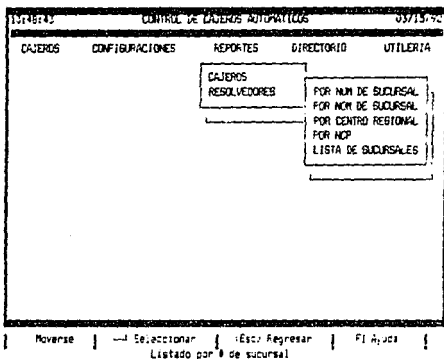
Al estar en la opción REPORTES y pulsar Enter se abre un submenú que presenta los diferentes tipos de reportes que se pueden generar.

13146743	CONTROL DE CAJEROS AUTOMÁTICOS			03/15/92
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA
CAJEROS RESOLVEDORES				

| Moveuse | — Seleccionar | <Esc> Regresar | F1 Ayuda |
Impresion de listados de cajeros.

REPORTES DE CAJEROS.

En la opción de CAJEROS del submenú REPORTES se abre otra ventana en la que se presentan los diferentes reportes que se pueden generar con la información de cajeros automáticos.



REPORTE POR NUMERO DE SUCURSAL.

Se genera un listado de las sucursales ordenado por número de sucursal. Si la impresora no está lista para imprimir se envía un mensaje de error.

09749752	CONTROL DE CAJEROS AUTOMATICOS			03/28/92
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA
		CAJEROS RESOLVEDORES		POR NUM DE SUCURSAL POR NOM DE SUCURSAL POR CENTRO REGIONAL POR MCP LISTA DE SUCURSALES
Novorse	— Seleccionar	(Esc) Regresar	Fl Ayuda	
	Impresora no esta lista.	— Continuar		

En caso contrario se inicia la impresión del reporte.

13752750	CONTROL DE CAJEROS AUTOMATICOS			03/15/92
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA
		CAJEROS RESOLVEDORES		POR NUM DE SUCURSAL POR NOM DE SUCURSAL POR CENTRO REGIONAL POR MCP LISTA DE SUCURSALES
Novorse	— Seleccionar	(Esc) Regresar	Fl Ayuda	
	Impresionando...			

REPORTE POR NOMBRE DE SUCURBAL.

Se genera un listado de las sucursales ordenado por nombre de sucursal. Si la impresora no está lista para imprimir se envía un mensaje de error.

En caso contrario se inicia la impresión del reporte.

REPORTE POR CENTRO REGIONAL.

Se genera un listado de las sucursales ordenado por número de centro regional. Si la impresora no está lista para imprimir se envía un mensaje de error.

En caso contrario se inicia la impresión del reporte.

REPORTE POR NCP.

Se genera un listado de las sucursales ordenado por NCP. Si la impresora no está lista para imprimir se envía un mensaje de error.

En caso contrario se inicia la impresión del reporte.

LISTA DE SUCURSALES.

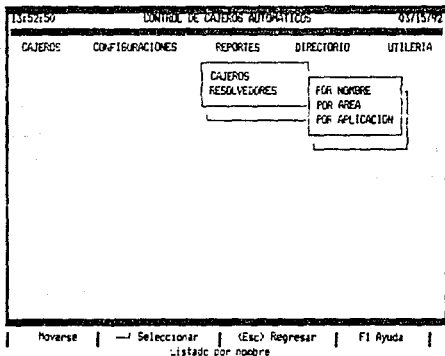
Se genera un listado con los datos de las sucursales (nombre, dirección, horario) ordenado por número de sucursal. Si la impresora no está lista para imprimir se envía un mensaje de error.

En caso contrario se inicia la impresión del reporte.

En estos últimos cuatro reportes las pantallas que se presentan son similares a las mostradas en el primer reporte.

REPORTES DE RESOLVEDORES.

En la opción de RESOLVEDORES del submenú REPORTES se abre otra ventana en la que se presentan los diferentes reportes que se pueden generar con la información del directorio.



REPORTE POR NOMBRE DE RESOLVEDOR.

Se genera un listado de los resolvedores ordenado por número de resolvedor. Si la impresora no está lista para imprimir se envía un mensaje de error.

09149152	CONTROL DE CAJEROS AUTOMATICOS				05/20/92
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA	
		CAJEROS RESOLVEDORES	POR NOMBRE POR AREA POR APLICACION		
Mostrar	— Seleccionar	(Esc) Regresar	FI Ayuda		
	Impresora no esta lista.	— Continuar			

En caso contrario se inicia la impresión del reporte.

11152150	CONTROL DE CAJEROS AUTOMATICOS				05/15/92
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA	
		CAJEROS RESOLVEDORES	POR NOMBRE POR AREA POR APLICACION		
Mostrar	— Seleccionar	(Esc) Regresar	FI Ayuda		
	Impresionando...				

REPORTE POR AREA.

Se genera un listado de los resolvedores ordenado por área. Si la impresora no está lista para imprimir se envía un mensaje de error.

En caso contrario se inicia la impresión del reporte.

REPORTE POR APLICACION.

Se genera un listado de los resolvedores ordenado por aplicación. Si la impresora no está lista para imprimir se envía un mensaje de error.

En caso contrario se inicia la impresión del reporte.

Las pantallas que aparecen en estos reportes son similares a las del reporte anterior.

DIRECTORIO

Al estar en la opción DIRECTORIO del menú principal y pulsar Enter se abre un submenú que presenta las operaciones que se pueden realizar con la información del directorio de resolvedores.

CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA
ALTA DE RESOLVEDOR BAJA DE RESOLVEDOR MODIF DE RESOLVEDOR CONSULTA DE RESOLVEDOR				

Reverse | ← Seleccionar | (Esc) Regresar | F1 Ayuda |
Registrar datos nuevos

Usando las flechas nos podemos mover a las distintas opciones y al pulsar Enter se selecciona alguna de ellas. F1 proporciona ayuda acerca de lo que se puede hacer en este nivel. Si se pulsa Esc regresamos al nivel anterior (menú principal).

ALTA DE RESOLVEDORES.

En esta opción se realiza el registro de la información de nuevos resolvedores (asesores) relacionados con el servicio de cajeros automáticos.

Al entrar a esta opción aparece la siguiente pantalla.

```
1375:150 CONTROL DE CAJEROS AUTOMATICOS 03/15/92
CAJEROS CONFIGURACIONES REPORTES DIRECTORIO UTILERIA
ALTA DE RESOLVEDOR
BAJA DE RESOLVEDOR
MODIF DE RESOLVEDOR
CONSULTA DE RESOLVEDOR
CLAVE 1030 NOMBRE HERNANDO HERNANDEZ DIAZ TEL 67907509
AREA 1ASTELIN APLICACION 1AUTOMATE EXTENSION 14316
Moverse | ← Seleccionar | Esc: Repasar | F1 Ayuda |
          | → Registrar | (Esc) Corregir |
```

Aquí se debe proporcionar la siguiente información:

Clave Número con el que se identifica al resolvedor, el cual consta de tres dígitos numéricos.

Nombre Nombre del resolvedor. Es un campo alfanumérico de veinte dígitos.

Teléfono Número telefónico de siete dígitos.

Area Departamento al que se encuentra asignado.

Aplicación Aplicación(es) que maneja. Campo alfanumérico de diez dígitos.

Extensión Extensión telefónica. Campo numérico de diez dígitos.

Después de dar los datos anteriores el sistema se encarga de validarlos y en el caso de que algún dato sea inválido se envía un mensaje de error y se regresará a corregir. Si los datos fueron correctos el sistema envía un mensaje para confirmar el registro de la información, en caso contrario se regresa a realizar las correcciones.

BAJA DE RESOLVEDORES.

En esta opción se realiza la baja de información de resolvedores. Al entrar a esta opción aparece la siguiente pantalla.

13/02/99	CONTROL DE CAJEROS AUTOMÁTICOS	07/15/92		
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA
ALTA DE RESOLVEDOR BAJA DE RESOLVEDOR MODIF DE RESOLVEDOR CONSULTA DE RESOLVEDOR				
CLAVE : APLICACION :				

| Moverse | ← Seleccionar | (Esc) Regresar | F1 Ayuda |

El sistema solicita la clave y la aplicación para proceder a la baja. Si los datos no son correctos se envía un mensaje de error. Si los datos son correctos se despliega la información relacionada y un mensaje para confirmar el borrado de la información, como se muestra a continuación.

10:52:50	CONTROL DE CAJEROS AUTOMÁTICOS				07/15/92			
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA				
<table border="1"><tr><td>ALTA DE RESOLVEDOR</td></tr><tr><td>BAJA DE RESOLVEDOR</td></tr><tr><td>MODIF DE RESOLVEDOR</td></tr><tr><td>CONSULTA DE RESOLVEDOR</td></tr></table>					ALTA DE RESOLVEDOR	BAJA DE RESOLVEDOR	MODIF DE RESOLVEDOR	CONSULTA DE RESOLVEDOR
ALTA DE RESOLVEDOR								
BAJA DE RESOLVEDOR								
MODIF DE RESOLVEDOR								
CONSULTA DE RESOLVEDOR								
<table border="1"><tr><td>CLAVE :001 APLICACION :CICSD07 AREA :ASTE</td></tr><tr><td>NOMBRE :HUBERT SIERRA TEL : 6563000 EXT :4317-4318</td></tr></table>					CLAVE :001 APLICACION :CICSD07 AREA :ASTE	NOMBRE :HUBERT SIERRA TEL : 6563000 EXT :4317-4318		
CLAVE :001 APLICACION :CICSD07 AREA :ASTE								
NOMBRE :HUBERT SIERRA TEL : 6563000 EXT :4317-4318								
<table border="1"><tr><td> </td></tr></table>								
Mostrar	Seleccíonar	(Esc) Regresar	F1 Ayuda					
Confirmar	Eliminar	(Esc) Corregir						

MODIFICACION DE RESOLVEDORES.

En esta opción se realiza la modificación de información de resolvedores registrados en el directorio.

Al entrar a esta opción aparece la siguiente pantalla.

13152750	CONTROL DE CARGA AUTOMÁTICO	03/15/92						
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA				
<table border="1"> <tr> <td>ALTA DE RESOLVEDOR</td> </tr> <tr> <td>BAJA DE RESOLVEDOR</td> </tr> <tr> <td>MODIF DE RESOLVEDOR</td> </tr> <tr> <td>CONSULTA DE RESOLVEDOR</td> </tr> </table>					ALTA DE RESOLVEDOR	BAJA DE RESOLVEDOR	MODIF DE RESOLVEDOR	CONSULTA DE RESOLVEDOR
ALTA DE RESOLVEDOR								
BAJA DE RESOLVEDOR								
MODIF DE RESOLVEDOR								
CONSULTA DE RESOLVEDOR								
<table border="1"> <tr> <td>CLAVE :</td> </tr> </table>					CLAVE :			
CLAVE :								
Moverse	— Seleccinar	Escr Regresar	Fl Ayuda					

El sistema solicita la clave del resolvedor al cual se van a realizar modificaciones. Si no se proporciona este dato entonces se solicita el nombre. Si no se proporciona alguno de estos dos datos o fueran incorrectos se envía un mensaje de error.

Si alguno de los datos es correcto se presenta la información para ser modificada.

13:52:50	CONTROL DE CAJEROS AUTOMATICOS			03/15/92																
CAJEROS	CONFIGURACIONES	FEFORTES	DIRECTORIO	UTILERIA																
<table border="1"> <tr> <td>ALTA DE RESOLVEDOR</td> </tr> <tr> <td>BAJA DE RESOLVEDOR</td> </tr> <tr> <td>MODIF DE RESOLVEDOR</td> </tr> <tr> <td>CONSULTA DE RESOLVEDOR</td> </tr> </table>					ALTA DE RESOLVEDOR	BAJA DE RESOLVEDOR	MODIF DE RESOLVEDOR	CONSULTA DE RESOLVEDOR												
ALTA DE RESOLVEDOR																				
BAJA DE RESOLVEDOR																				
MODIF DE RESOLVEDOR																				
CONSULTA DE RESOLVEDOR																				
<table border="1"> <tr> <td>CLAVE : 001 NOMBRE : HUEFT SIERRA GONZALEZ</td> <td>TEL : 7304589</td> </tr> <tr> <td>AREA : ASTELIN</td> <td>EXTENSION : 4218</td> </tr> </table>					CLAVE : 001 NOMBRE : HUEFT SIERRA GONZALEZ	TEL : 7304589	AREA : ASTELIN	EXTENSION : 4218												
CLAVE : 001 NOMBRE : HUEFT SIERRA GONZALEZ	TEL : 7304589																			
AREA : ASTELIN	EXTENSION : 4218																			
<table border="0"> <tr> <td>Moverse</td> <td> </td> <td>← Selecionar</td> <td> </td> <td>(Esc) Regresar</td> <td> </td> <td>F1 Ayuda</td> <td> </td> </tr> <tr> <td>Area no registrada.</td> <td> </td> <td>→ Registrar</td> <td> </td> <td>(Esc) Corregir</td> <td> </td> <td></td> <td> </td> </tr> </table>					Moverse		← Selecionar		(Esc) Regresar		F1 Ayuda		Area no registrada.		→ Registrar		(Esc) Corregir			
Moverse		← Selecionar		(Esc) Regresar		F1 Ayuda														
Area no registrada.		→ Registrar		(Esc) Corregir																

Después de realizar las modificaciones se envía un mensaje para confirmar el registro de las correcciones.

CONSULTA DE RESOLVEDORES.

En esta opción se realiza la consulta de información de resolvedores registrados en el directorio.

Al entrar a esta opción aparece la siguiente pantalla.

13132150	CONTROL DE CAJEROS AUTOMATICOS	03/15/92			
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA	
<table border="1"> <tr> <td> ALTA DE RESOLVEDOR BAJA DE RESOLVEDOR MODIF DE RESOLVEDOR CONSULTA DE RESOLVEDOR </td> </tr> </table>					ALTA DE RESOLVEDOR BAJA DE RESOLVEDOR MODIF DE RESOLVEDOR CONSULTA DE RESOLVEDOR
ALTA DE RESOLVEDOR BAJA DE RESOLVEDOR MODIF DE RESOLVEDOR CONSULTA DE RESOLVEDOR					
<table border="1"> <tr> <td> CLAVE : </td> </tr> </table>					CLAVE :
CLAVE :					
Noverse	Seleccinar	Escr Regresar	F1 Ayuda		

El sistema solicita la clave del resolvedor al cual se va a realizar consultas. Si no se proporciona este dato entonces se solicita el nombre. Si no se proporciona alguno de estos dos datos o fueran incorrectos se envia un mensaje de error.

Si alguno de los datos es correcto se presenta la información para su consulta.

13752150	CENTRO DE CAJEROS AUTOMATICOS			03715792				
CAJEROS	CONFIGURACIONES	REPORTES	DIRECTORIO	UTILERIA				
<table border="1"> <tr> <td>ALTA DE RESOLVEDOR</td> </tr> <tr> <td>BAJA DE RESOLVEDOR</td> </tr> <tr> <td>MODIF DE RESOLVEDOR</td> </tr> <tr> <td>CONSULTA DE RESOLVEDOR</td> </tr> </table>					ALTA DE RESOLVEDOR	BAJA DE RESOLVEDOR	MODIF DE RESOLVEDOR	CONSULTA DE RESOLVEDOR
ALTA DE RESOLVEDOR								
BAJA DE RESOLVEDOR								
MODIF DE RESOLVEDOR								
CONSULTA DE RESOLVEDOR								
<table border="1"> <tr> <td>CLAVE 1002 NOMBRE : GUADALUPE AGUILAR</td> <td>TEL : 7903212</td> </tr> <tr> <td>AREA : RSTE</td> <td>EXTENSION : 4317-4318 APLICACION : CIGSDAT</td> </tr> </table>					CLAVE 1002 NOMBRE : GUADALUPE AGUILAR	TEL : 7903212	AREA : RSTE	EXTENSION : 4317-4318 APLICACION : CIGSDAT
CLAVE 1002 NOMBRE : GUADALUPE AGUILAR	TEL : 7903212							
AREA : RSTE	EXTENSION : 4317-4318 APLICACION : CIGSDAT							

Moverse | ← Seleccionar | (Esc) Regresar | F1 Ayuda |
 — Continuar | (Esc) Terminar

La tecla ESC, al igual que F1, puede ser presionada en cualquier momento y su efecto será el de regresarnos al nivel anterior.

SALIDA DEL SISTEMA.

Para terminar una sesión de trabajo basta con estar en el menú principal y presionar la tecla ESC para regresar al nivel del sistema operativo.

CONCLUSIONES

La instalación y uso de este sistema en el área de teleproceso aporta beneficios tanto a nivel departamento como a nivel personal.

A nivel departamento se logra una independencia de información respecto a otras áreas, ya que se cuenta con un medio propio que facilita el desempeño de las funciones de diagnóstico y corrección de fallas, a la vez que permite contar con información actualizada.

A nivel personal, la consulta de información se vuelve más ágil al contar con un estándar para la presentación de los datos y la obtención de reportes de se vuelve más variada y, en general, se consigue un mejor aprovechamiento de la información.

Cabe señalar que este sistema puede ser aprovechado por otras áreas involucradas en el servicio de cajeros, ya que sus requerimientos de información son similares a los a del área de teleproceso.

Por otro lado, se logra hacer de la PC una herramienta más para el funcionamiento del área, ya que se involucra en la creación de los formatos y configuraciones que anteriormente se tenían que realizar utilizando medios no adecuados.

Si bien este sistema aporta los beneficios mencionados, también, como todo sistema, es susceptible de ser adecuado a nuevos requerimientos.

Entre las mejoras que se pueden incorporar al sistema está la seguridad para el acceso, detalles en la presentación del sistema, variantes en la generación de reportes y consulta de información. Asimismo, se puede desarrollar un módulo para el manejo de gráficos que explote la información almacenada en la base de datos.

En general, se satisfacen los requerimientos planeados por el área de teleproceso y se cumple con el objetivo establecido para el desarrollo de este sistema.

GLOSARIO.

ATM	_____	Automatic Teller Machine. (Cajero automático).
BIOS	_____	Basic Input Output System.
DBMS	_____	Data Base Manager System. (sistema manejador de bases de datos).
EFTS	_____	Electronic Fonds Transfer System. (sistema de transferencia electrónica de fondos).
Hardware	_____	Equipo electrónico y electromecánico.
HOST	_____	Huésped. (computador central).
LU	_____	Logical Unit. (unidad lógica).
MS-DOS	_____	MicroSoft-Disk Operating System. (sistema operativo para computadora personal).
NAU	_____	Network Addressable Unit. (unidad direccionable de la red).
NCP	_____	Network Control Program. (Programa para el control de la red).
Nodo	_____	Equipo de comunicaciones capaz de administrar recursos de la red.
PC	_____	Personal Computer. (computadora personal).

PU _____ Phisycal Unit. (unidad física).

RAM _____ Random Access Memory. (memoria de acceso aleatorio).

RS232C _____ Protocolo de comunicaciones para interconexión de equipo.

SNA _____ System Network Architecture. (arquitectura para sistemas de red).

Software _____ Conjunto de programas para manejar datos y comportamiento de equipo de cómputo.

SSCP _____ System Services Control Point. (punto de control para los servicios del sistema).

VGA _____ Video Graphics Adapter. (adaptador para video gráfico).

VTAM _____ Virtual Telecommunications Access Method (método de acceso virtual a telecomunicaciones).

BIBLIOGRAFIA.

- Fundamentos de bases de datos. Henry F. Korth - Abraham Silberschatz.
Ed. Mc. Graw-Hill. 1990.
- Análisis y diseño de sistemas de información.
James A. Senn.
Ed. Mc. Graw-Hill.
- Bases de Datos.
C.J. Date.
Ed. Sitesa.
- Organización de bases de datos.
Jame Martin.
Ed. Prentice Hall.
- Diseño de bases de datos.
Wiederhold.
Mc. Graw-Hill.
- Ingeniería de software. Richard Fairley.
Ed. Mc. Graw-Hill. 1987.
- Ingeniería de sistemas. Victor Flores Zavala.
Facultad de Ingeniería. 1982.
- Comunicaciones, interfases, modems, protocolos, redes y normas.
José M. Huldobro.
Ed. Faraninfo.
- Introducción a la teoría de sistemas de comunicaciones.
P.B. Lathi.
Ed. Limusa. 1980.
- El libro del CLIPPER SUMMER 87. Antonio Quiroz Casado -Antonio Torres Lozano.
Ed. Macrobit Editores. 1991.
- CLIPPER 5.0 : Guía del compilador para dBase III Plus y Dbase IV.
García - Badell.
Ed. Mc. Graw-Hill. 1991.

-Illustrated CLIPPER 5.0
Wordware publishing Inc.

-Clipper. Teoría, aplicaciones y rutinas de programación.
Ed. Macrobit.

-dBase III Plus.
J.M. Alonso García del Busto.
Ed. Paraninfo. 1988.

-Lotus 1-2-3. Guía del usuario.
Edward M. Baras.
Ed. Mc. Graw-Hill. 1989.

-DB2/SQL Manual para programadores.
Tim Martyn - Tim Hartley.
Ed. Mc Graw-Hill.

-Sistema/88. Manual de operación.
Bancomer S.A.

-Sistemas de comunicaciones para aplicaciones bancarias.
Bancomer S.A.

-Fundamentos de SNA.
Bancomer S.A.

-Mundo Bancomer.
Bancomer S.A.

-Time express. Vol. 4 Núm. 5. Septiembre 1988.

-Time express. Vol. 4 Núm 7. Septiembre 1988.

-BYTE, septiembre de 1990.

-PC Magazine. Vol. 1 Número 2. 1990.